ファミリー コンピュータ™専用

FAMILY BASIC™

ファミリー ベーシック™：形式 HVC-BS

取扱説明書

ゲームBASIC

カリキュレータボード

ミュージックボード

メッセージボード



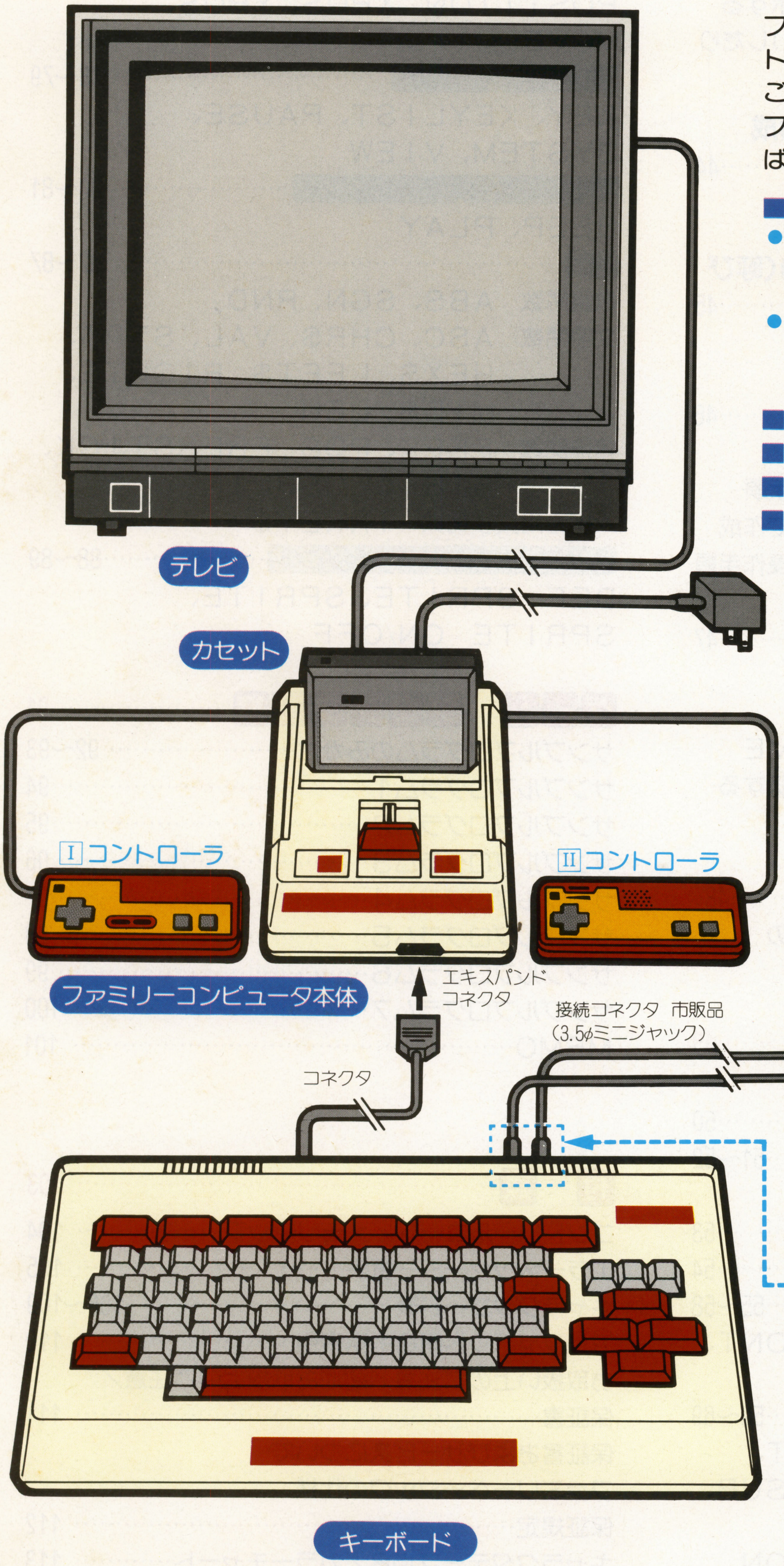
Nintendo®

ごあいさつ

このたびは任天堂〝ファミリーベーシック（HVC−BS）〟をお買いあげいただきまして、誠にありがとうございました。
ご使用の前に取扱い方、使用上の注意等、この「取扱説明書」をよくお読みいただき、正しい使用法でご愛用ください。
なお、この「取扱説明書」は大切に保管してください。

もくじ

ファミリーベーシック・システム構成図



ファミリーベーシック（キーボード・カセット）は、ファミリーコンピュータに接続してご使用ください。
ファミリーコンピュータの世界を楽しさいっぱいに拡げます。

■GAME BASICモード

●BASIC

（マリオやレディなどのアニメキャラクタを画面に登場させ、動かすプログラムを作ります。）

●BG GRAPHIC

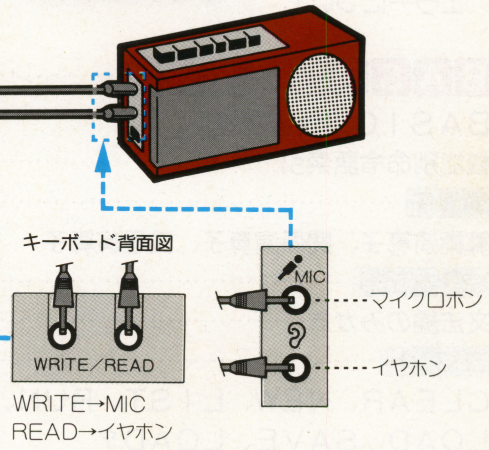
（アニメキャラクタの背景画を作ったり、背景用キャラクタを使って絵を描くことができます。）

■カリキュレータボード（計算）
■ミュージックボード（音楽演奏）
■メッセージボード（メモ・伝言）
■コンピュータのうらない

●コントローラでゲーム

GAME BASICでは、オリジナルゲームをつくれます。コントローラを利用するプログラムにすれば、2人でゲームが楽しめます。

モノラルラジカセ（シャープQT-10等 市販品）



●SAVE（保存）、LOAD（呼び出し）

キーボードには、モノラルラジカセを利用して、ゲームのプログラムやBG GRAPHICの画面で描いた絵を、SAVE（保存）、LOAD（呼び出し）するための端子がついています。
大切なオリジナルゲームは、カセットテープにSAVEしておけば、安心です。いつでもLOADして、楽しむことができます。
（詳しくは45ページをご覧ください。）

●多機能キーボード

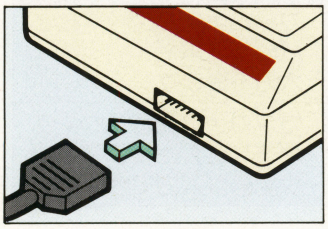
数字や英文字、カタカナはもちろん特殊な記号が入力できます。カタカナは、アイウエオ順に並んでいますので使いやすくなっています。



電源を入れ、コンピュータを呼び出すには……

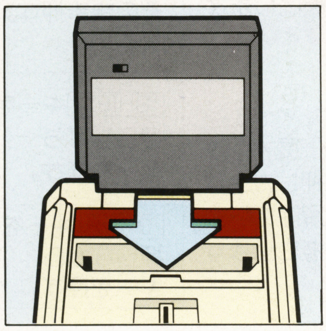
1 キーボードのコネクタを差し込む

●電源(POWER)をOFFにしたファミリーコンピュータのエキスパンドコネクタ(機能拡張コネクタ)にキーボードのコネクタを接続します。



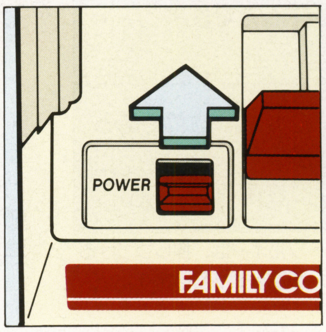
2 カセットを挿入する

●カセットをファミリーコンピュータのカセットコネクタに挿入します。



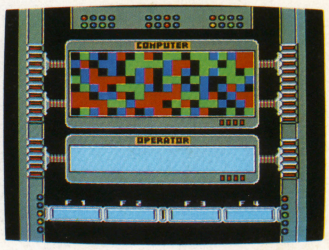
3 電源をONにする

●ファミリーコンピュータの電源(POWER)を入れます。



4 スタート画面

●スタート画面でコンピュータのカラーパネルが表われます。



メモリバックアップについて……………

(カセットを本体から抜くときは必ずONにし、次にコンピュータからOFFの指示が出るまで、そのままの状態にしておいてください。)

単三乾電池を利用して、カリキュレータ・ミュージック・メッセージの各ボードで入力したデータやBASICプログラムを、一時的にカセット内に記憶(メモリバックアップ)しておくことができます。

●バックアップスイッチのON、OFFはコンピュータの指示に従ってください。
●ただし、GAME　BASICモードの場合に限り「データヲ　ノコシタイトキハ　カセットノ　バックアップスイッチヲ　ONニ　シテクダサイ」の指示は表示されません。
BASICのプログラムをメモリバックアップする場合は、GAME　BASICモードを終了してスタート画面になったとき、バックアップスイッチをONにしてください。
●コンピュータの指示に従わずにバックアップスイッチをON・OFFすると、正確なデータが記憶(メモリバックアップ)されなかったり、正常動作しない場合がありますのでご注意ください。
●BASICプログラムのデータは、カリキュレータ・ミュージック・メッセージの各ボードを実行すると消えてしまいます。
また、カリキュレータ・ミュージック・メッセージの各ボード画面のデータは、BASICモードを実行すると消えます。
●メモリバックアップしたデータを利用される場合は、上記の点を十分にご注意ください。

カセット(表)





カセット(裏)

電池交換について………………………

●カセットにはメモリバックアップのために単三乾電池2本を使用します。
バックアップスイッチをONの状態にしたとき、約2ヵ月(単三乾電池SUM−3(E)使用時)です。OFFにすると電気の消費量が大きくなり、寿命は短くなってしまいます。
●カセットをファミリーコンピュータに差し込んだ状態(本体の電源はON)で電池を交換すると、記憶されていたデータやプログラムは残っていますが、カセットを抜いた状態で交換すると消えてしまいます。
●111ページの「乾電池お取扱い上のご注意」をご覧ください。

BASICカセットでメモリバックアップできるもの…………

●BASICプログラム
●カリキュレータボード画面のデータ
●ミュージックボード画面のデータ
●メッセージボード画面のデータ

(ご注意)BG GRAPHICの画面のデータはメモリバックアップできません。

カセットテープでSAVE・LOADできるもの……

●BASICプログラム
●BG GRAPHICの画面のデータ

７２キーの本格的キーボード。
カタカナは使いやすい50音配列です。

文字キー

数字・英文字・カタカナ・記号を入力するためのキーです。
キーにかかれてある上下左右の文字記号を次のルールで入力します。

英・数字状態	●カーソル：□ ●羽根ペン：オレンジ
キーをそのまま押す。	キーの左にかかれてある数字と英文字(大文字)や記号が入力できます。
SHIFT キーを押しながらキーを押す。	キーの上にかかれてある記号が入力できます。

※文字キー、スペースキー、カーソルキーは、押しつづけると連続して入力できるリピート機能をもっています。

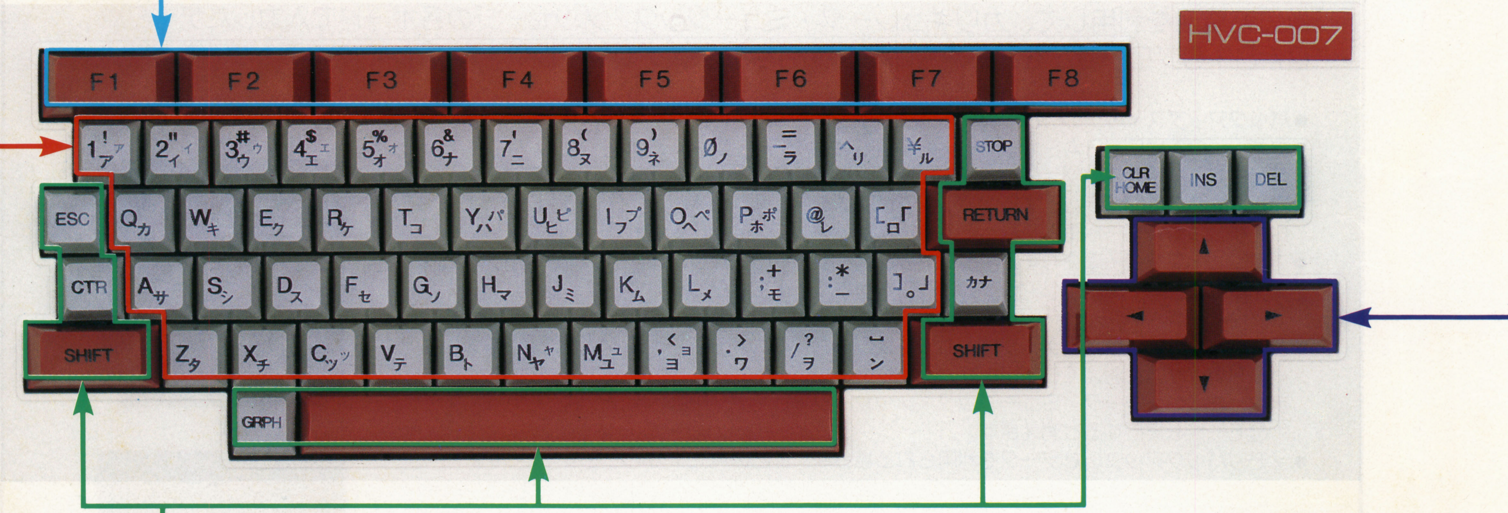
カナ キーを押しカナ状態にします

もう一度 カナ キーを押しカナ状態を解除、英・数字状態にします

カナ状態	●カーソル：― ●羽根ペン：グリーン
キーをそのまま押す。	キーの下にかかれてあるカタカナ(大文字)や記号が入力できます。
SHIFT キーを押しながらキーを押す。	キーの右にかかれてあるカタカナ(小文字)や記号、パピプペポが入力できます。
GRPH キーを押しながらキーを押す。	カタカナの濁音（ガギグゲゴなど）とグラフィック記号が入力できます。

ファンクションキー

それぞれの画面で違った役割をもつキーです。
(ファンクションキーの詳しい説明は各ボード画面をご覧ください)



特殊キー

文字キーと組み合わせて使ったり、
画面上でいろいろな操作をするときに使うキーです。

ここに書いてある機能以外にもいろいろな働きがあります。各画面の説明やBASICのページをご覧ください。

ESC エスケープキー
- カリキュレータ、ミュージック、メッセージ各ボードからセレクト画面に戻るとき等に使います。

CTR コントロールキー
- GAME BASICモードで指定のキーと組み合わせて、いろいろなコントロールができます。

SHIFT シフトキー
- 文字キーと組み合わせて、カタカナの小文字やキーの上と右にかかれてある記号が入力できます。

GRPH グラフィックキー
- 文字キーと組み合わせて、カタカナの濁音が入力できます。
- グラフィック記号（文字キーの最上列のキー のみ）が入力できます。

［　　］スペースキー
- 文字と文字の間に１文字分の空白(スペース)をつくります。

※本文中では SPACE キーと表示しています。

STOP ストップキー
- BASICプログラムの実行を中断（BREAK）するときに使います。
- スタート画面、セレクト画面の表示を一時止める働きをします。
- BG GRAPHICの画面からGAME BASICモード画面に戻るときに使います。

カナ カナキー
- カタカナやキーの下や右にかかれてある記号を入力するときに使います。一度押すとカナ状態になり、もう一度押すと解除されます。

RETURN リターンキー
- 入力した文字や記号がメモリされると同時にカーソルが次の行の先頭に戻ります。
- 羽根ペンはインクつぼに入ってから改行します。

CLR HOME クリアホームキー
- そのまま押すと画面の表示は消えずにカーソルはホーム位置(画面左上)に戻ります。
- SHIFT キーを押しながら押すと画面の表示が消えて、カーソルがホーム位置に戻ります。



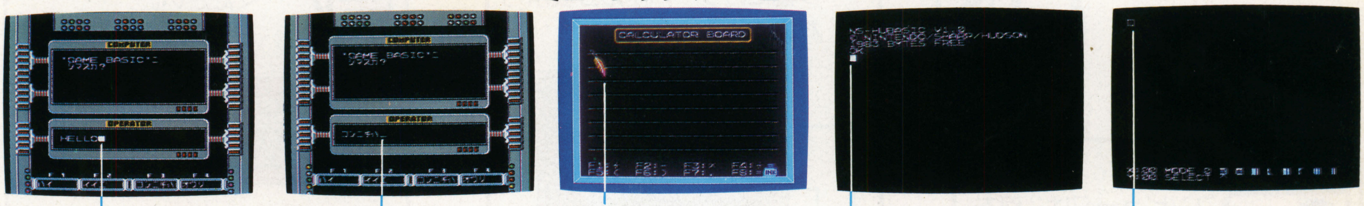
キーの位置は押してるうちに指がおぼえてくれる！
君のコトバをキーボードで伝えよう。

キーボード各キーの名称と使い方

カーソルキー

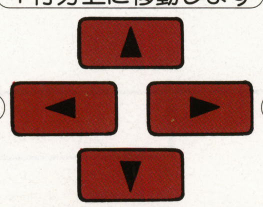
カーソルキーは文字や数字の入力する位置（カーソル）を上下左右に動かすキーです。

メニュー画面(英文字)　メニュー画面(カナ)　バイオリズムボード画面／カリキュレータボード画面／ミュージックボード画面　BASICの画面　BG GRAPHICの画面



カーソル　羽根ペン　カーソル

それぞれのカーソルキーを１回押すごとに
カーソル／羽根ペンが…



各ボード画面とBASICで部分修正をするときのカーソルキーの使い方。

●すでに入力した文字にカーソルを移すとその文字を自由に修正できます。

トマダ　ハナコ_	"ト"を"ヤ"に修正するとき…
トマダ　ハナコ	カーソルを"ト"に移します。
ヤマダ　ハナコ	ヤを入力します。
ヤマダ　ハナコ_	カーソルをもとの位置に戻します。

※BASICでは、修正後にRETURNキーを押してください。

INS.キー・DELキーの使い方

INSキー

●あとから文字をわり込ませるにはINSキーを使います。

ACDEF■	"A"と"C"のあいだに空白をつくり"B"をわり込ませるとき…
A■DEF	カーソルを"C"に移します。
A■CDEF	INSキーを押すと"A"と"C"のあいだに１文字分の空白ができます。
AB■DEF	Bを入力します。
ABCDEF■	カーソルをもとの位置に戻します。

※BASICでは、修正後にRETURNキーを押してください。

DELキー

●余分な文字を消すにはDELキーを使います。

ABCXDE■	"X"を消すとき……
ABCX■E	消したい文字の**右どなり**にカーソルを移します。
ABC■E	DELキーを押すと"X"が消え、"D"より右の文字がつまります。
ABCDE■	カーソルをもとの位置に戻します。

※BASICでは、修正後にRETURNキーを押してください。

INSインサートキー
●カーソルの位置に空白をつくります。(詳しくは右の説明をご覧ください。)

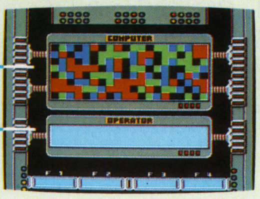
DELデリートキー
●カーソルの左の位置にある文字を消去します。(詳しくは右の説明をご覧ください。)

コンピュータが話しかけてきたら自己紹介しよう。

POWER ON

スタート画面

1 スタート画面があらわれて、コンピュータからのメッセージ欄にカラーパネルが表示されます。



あなたからのメッセージ欄。
コンピュータからのメッセージ欄。

2 キーボードのキーをどれかひとつ押してください。



文字キーのひとつか
[SPACE]キーを押してください。

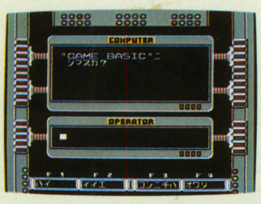
[3]キーを押す

※カセットのバックアップスイッチの操作指示は、画面に表示されませんので、データを残したい時はスタート画面に戻ってからONにしてください。

セレクト画面

6 コンピュータは、何を実行するか順にたずねてきます。選んでください。

- GAME BASIC
- カリキュレータボード
- ミュージックボード
- メッセージボード



- [ハ][イ][RETURN]キー、または[F1][RETURN]キーを押す。
- [イ][イ][エ][RETURN]キー、または[F2][RETURN]キーを押す。
- コンピュータにあいさつすると、うらないをしてくれます。(詳しくはNo.8をご覧ください。)
- [オ][ワ][リ][RETURN]キー、または[F4][RETURN]キーを押すと、コンピュータは動作を終了します。

サギョウヲ チュウシ シマス

データヲ ノコシタイ トキハ
カセットノ バックアップスイッチヲ
ONニ シテクダサイ

マリオ
サン マタ ヨビダシテ クダサイ

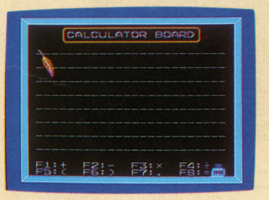
サヨウナラ!

このメッセージ表示後、スタート画面に戻ります。

各ボードの実行画面

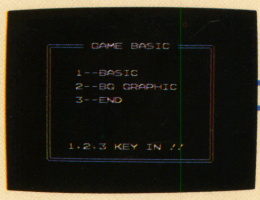
7 それぞれの実行画面にかわります。

● カリキュレータボード画面



(操作方法→8ページ)

● GAME BASICモード画面



[1]キーか[2]キーを押してモードを選びます。

[1]キー

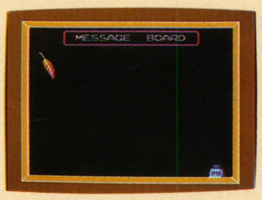
[2]キー

● ミュージックボード画面



(操作方法→10ページ)

● メッセージボード画面



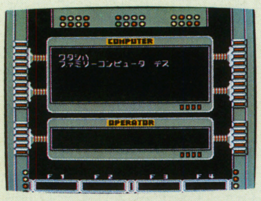
(操作方法→9ページ)

[ESC]キーを押す



コンピュータと対話しながら
多彩なメニューが楽しめる!

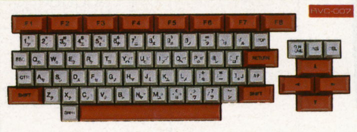
3 カラーパネルの点滅が消え、コンピュータが話しかけてきます。



ドウサヲ カイシ シマス

4 あなたの名前を入力して RETURN キーを押してください。

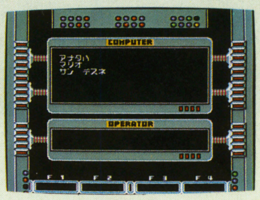
アナタハ ダレデスカ?
ナマエヲ イレテ クダサイ



名前は英文字でもカタカナでもかまいません。18文字まで入力できます。カタカナの場合は、カナ キーを押してから入力してください。
※何も入力しない場合は、そのまま RETURN キーを押してください。

5 コンピュータがあなたの名前を確認して作業を開始します。

リョウカイ!



サギョウヲ カイシ シマス

4でマリオと入力した場合。

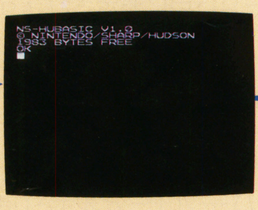
ESC キーを押した後、STOP キを押す

SYSTEM RETURN キーを押す

を押す

を押す

〈BASIC〉



〈BG GRAPHIC〉



8 コンピュータのうらない

- HELLO RETURN
- オハヨウ RETURN
- コンニチハ RETURN
- コンバンハ RETURN
- F3 RETURN

を入力すると、コンピュータはあいさつを返してきます。その後、うらないをしてくれます。

マリオ
サン ウラナイヲ シマスカ?

- **ハイかイイエで答えます。**

ハイ RETURN キー、
または F1 RETURN キーを押す

ツギノ シツモンニ コタエテクダサイ

セイネンガッピヲ イレテクダサイ

(操作方法→9ページ)

イイエ RETURN キー、
または F2 RETURN キーを押すと、セレクト画面に戻る。

9 コンピュータの理解できる言葉

- ハイ RETURN
- イイエ RETURN

} コンピュータの質問に答えます。

- HELLO RETURN
- オハヨウ RETURN
- コンニチワ RETURN
- コンバンワ RETURN

} セレクト画面でコンピュータにあいさつができます。

- END RETURN
- オワリ RETURN

} コンピュータは動作を終了します。

- BASIC RETURN
  セレクト画面で、GAME BASIC画面を直接選ぶことができます。
- CAL. RETURN
  セレクト画面でカリキュレータボード画面を直接選ぶことができます。
- MUS. RETURN
  セレクト画面でミュージックボード画面を直接選ぶことができます。
- MES. RETURN
  セレクト画面でメッセージボード画面を直接選ぶことができます。

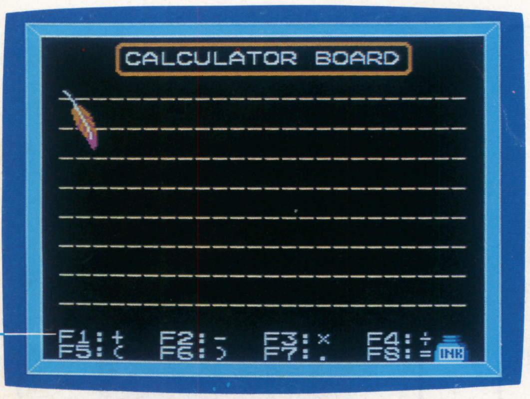
※コンピュータの問いかけに返答しない(約90秒)と、自動的に作業を中止し、スタート画面に戻ります。

カリキュレータボード画面

カリキュレータボード画面の操作方法

計算式を入力すれば、コンピュータが答える。
画面をフルに使える計算ボード！





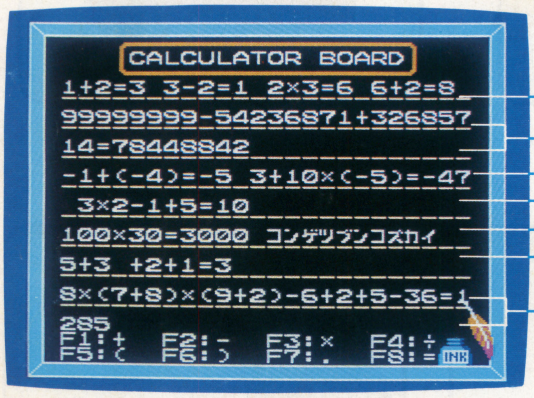
ファンクション表示

羽根ペンが 動いて計算式を表示します。
ヨコ24文字×タテ８行を使って計算ができます。

カリキュレータボード基本ルール

計算式	通常の四則計算と（　）カッコ付の計算ができます。カッコ付の計算は１重カッコが１つの計算式の中で10個まで使えます。
有効数字	８桁までを計算します。８桁以下は切り捨てます。±(99999999～0.0000001)
符号	＋　－　×　÷　（　）．＝はファンクションキーで入力できます。
計算の優先順位	①（　）の中　②×または÷　③＋または－の順です。優先順位が同じときは、その行の先頭から計算します。

〈計算例〉

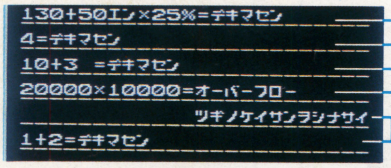


計算式の入力方法は…

- 文字キーとファンクションキーで数字や符号を入力すると、羽根ペンが式を書きます。
- ＝(イコール)を入力すると、答えが出ます。羽根ペンは答えの後に１文字分の空白(スペース)をあけて移動します。
- RETURN キーを押すと、羽根ペンは改行します。また、行の最後(24文字目)を入力すると、自動的に改行します。

A…１行でいくつもの計算ができます。
B…計算式は２行以上にわたって入力できます。
C…符号に続く－（マイナス）の数字はカッコでくくります。
D…計算式の前後には１文字分以上の空白が必要です。改行した場合も同様です。
E…メッセージも入力できます。
F…計算式の中に空白があるときは空白の前の計算はされません。
G…８行に答えが表示しきれない場合は、９行目に表示されます。

〈エラー表示〉



★次の場合にはエラーになります。

H…計算式の中に数字や符号以外の文字や記号を入力したとき
I…計算式になっていないものを＝(イコール)でつないだとき
J…＝(イコール)の前に空白があったとき
K…計算結果が８桁を越えたとき("オーバーフロー"と表示されます)
L…行の最後までメッセージを入力し、改行して空白をあけずに計算式を入力したとき

〈６を５に修正した場合〉



数式を部分修正するには…

- ▲▼◀▶キーで羽根ペンを動かし、すでに書き込んである数式を修正します。＝(イコール)は消えますから、再び＝（イコール)を入力してください。
- INS DEL キーで、すでに書き込んだ計算式の途中に、数字や符号を割り込ませたり、取り除いたりすることができます。
- 新しい答えが、うしろの式に重なった場合は、うしろの式が消去されます。

※左のように算数ドリルもつくれます。＝(イコール)を入力すると答えがでます。計算した結果が正しいかどうか確かめてください。

算数ドリルもつくれます。



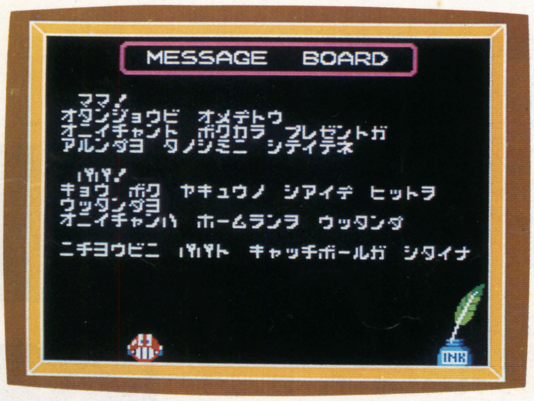
- 画面をすべて消去するには SHIFT キーを押しながら、CLR HOME キーを押します。
- カリキュレータボード画面を終了したいときは、ESC キーを押してください。

メッセージボード画面



メッセージボード画面の操作方法

文字・記号を入力すれば、マリオやレディ等いろいろなキャラクタ達が登場します。



羽根ペンが動いてメッセージを表示します。
(ヨコ24文字×17行)

メッセージの入力方法……

- 文字キーと[GRPH]キーで数字・英文字・カタカナ・グラフィック記号等を自由に入力することができ、羽根ペンがメッセージを書きます。
- [RETURN]キーを押すと改行します。また行の最後(24文字目)を入力すると、自動的に改行します。

部分修正するには……

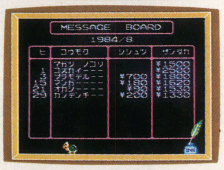
- [▲][▼][◀][▶]カーソルキーで訂正したい箇所に羽根ペンを移動させて修正してください。
- [INS][DEL]キーで、すでに書き込んだメッセージの途中に、文字や記号を割り込ませたり、取り除いたりすることができます。

> - 画面をすべて消去するには[SHIFT]キーを押しながら、[CLR HOME]キーを押します。
> - メッセージボード画面を終了したいときは、[ESC]キーを押してください。

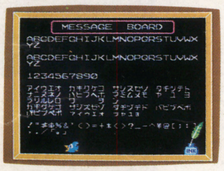
メモリバックアップを利用しましょう。



1.伝言板
「ママからパパへ」「パパからボクへ」「ボクからママへ」……いろいろなメッセージを書いて伝えてください。



2.手軽な記録板
「こずかい帳」「お買い物リスト」などメモがわりに使ってください。



3.文字の入力練習
このキーボードは、英文字・記号がタイプライターと同じキー配列になっています。早く正確に指を動かす練習ができます。

コンピュータのうらない

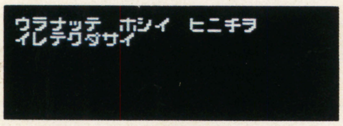


うらない画面の操作方法

コンピュータにあいさつをすると占いをしてくれます。







- [H][E][L][L][O][RETURN]
- [オ][ハ][ヨ][ウ][RETURN]
- [コ][ン][ニ][チ][ワ][RETURN]
- [コ][ン][バ][ン][ワ][RETURN]
- [F3][RETURN]

を入力すると、コンピュータもあいさつをし、占いをしてくれます。
- コンピュータが「ウラナイヲ　シマスカ?」とたずねてきたら[ハ][イ][RETURN]か[F1][RETURN]を入力してください。
- 生年月日と占ってほしい日を入力すると、その日のコンディションを予想し、また生まれてから何日経過したかを正確に教えてくれます。

年月日の入力方法……

- 西暦・年号は次のように入力します。

西暦→4桁を入力します。1984
年号→明治――M1~M44
　　　大正――T1~T14
　　　昭和――S1~S999

※明治45年は大正1年、大正15年は昭和1年としてください。
※年と月と日の間には、数字以外の文字を入れるか、あるいはスペースをあけてください。
〔例〕　1984/12/23
　　　　S59␣6␣23
- 占いのできる範囲は、1868年(明治元年)~2924年(昭和999年)の間です。

この占いはバイオリズムに基づいています。身体・感情・知性、それぞれに高調・低調・要注意日の判定をし、特に目立った結果を表示しています。

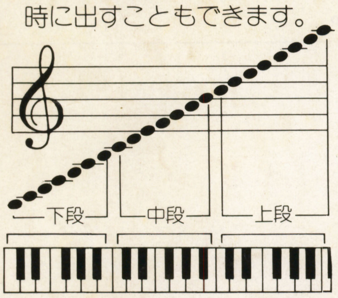
ミュージックボード画面

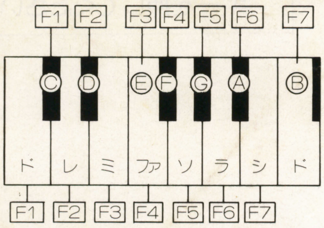


SHIFTキーを押しながらF8キーを押す：$(演奏のくり返しマークが入力されます。
F8キーを押す：演奏を開始します。

音階の入力欄

- 音階の入力欄は上段、中段、下段に分かれています。それぞれの段を利用して3オクターブの音を出すことができます。
- 入力欄は3音×24マス×4行です。上・中・下段の3音を同時に出すこともできます。





※F1~F7キーを押すと、ド~シが表示されて音が出ます。
※SHIFTキーを押しながらF1~F7キーを押すと、C~Bが表示され半音高い音が出ます。

音階の修正方法は……

- ▲▼◀▶カーソルキーで羽根ペンを訂正したい箇所に移動させて修正します。
- 演奏中に修正したい場合は、CLR HOMEキーを押して羽根ペンをインクつぼから画面左上に移します。▲▼◀▶キーで訂正したい箇所に羽根ペンを移動させて修正します。

演奏の楽しみ方は……

- F8キーを押すと、羽根ペンはインクつぼに入り、演奏が始まります。
- 上・中・下の音は同時に演奏され、カーソルが右へ移動して演奏している箇所を知らせます。
- 演奏中に◀▶キーを押すと、FAST<——□———>SLOWのカーソルが移動し、演奏する速さが変化します。

  ◀…速い　▶…遅い

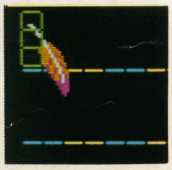
- SPACEキーを押すと、演奏は中止されます。◀▶キーを押すごとに演奏スケールは1マスずつ左右に移動します。演奏を再開するときは、F8キーを押してください。
- 演奏中にF8キーを押すと、曲の最初から演奏を始めます。

※IIコントローラーについているマイクロホンを利用すれば、カラオケで楽しむこともできます。

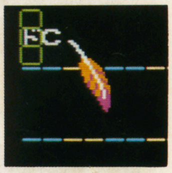
音階の入力方法は…

音階入力時には、演奏スケールは動きません。

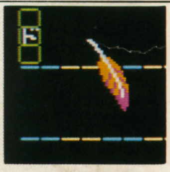
1. 羽根ペンは、1行目の上段、1音目にあります。▲▼◀▶キーで羽根ペンを入力したい位置に移動します。



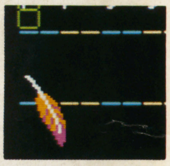
2. 位置を決めたらファンクションキーで音を入力します。
   - ファンクションキーF1~F7は、それぞれド~シに対応します。
   - SHIFTキーを押しながらファンクションキーF1~F7を押すと、それぞれ半音上がった音C~Bが入力されます。



3. 羽根ペンの位置に音が入力されると同時に羽根ペンは右へ1音分移動します。
   - 音階を入力しない場合は、SPACEキーで空白をつくります。



4. 1行目（24マス）の入力を終了すると、羽根ペンは次の行の1音目へ移動します。



5. くり返し演奏のマーク$を書きます。曲をくりかえし演奏させるときは、$マークを入力しておきます。上段・中段・下段のどの位置に入力してもかまいません。



＊文字キーを用いると、より複雑な演奏を楽しめます。

- 上段に数字の1234567を入力すると、中音域の音が出ます。
- 中段に1234567を入力すると高音域の音が出ます。
- 下段に1234567を入力すると低音域のさらに1オクターブ低い音が出ます。
- カタカナのアイウエオカキを入力すると、1~7までのそれぞれ半音高い音が出ます。

※文字キーを使うことによって、4オクターブの音を出すことができます。

- INS DELキーで、すでに書き込んだ音階の途中に、割り込ませたり、取り除いたりすることができます。

> - 画面をすべて消去するにはSHIFTキーを押しながら、CLR HOMEキーを押します。
> - ミュージックボード画面を終了したいときは、ESCキーを押してください。

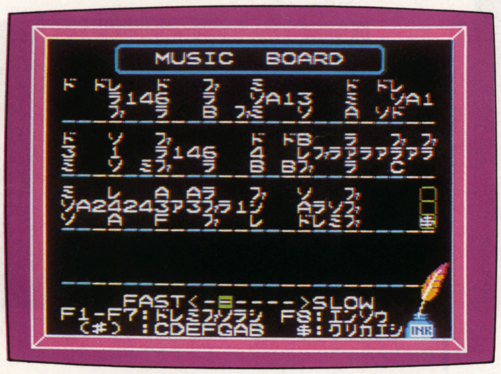
ミュージックボード画面の操作方法



音階を入力すればコンピュータが自動演奏。作曲もできる。

いろいろな曲の入力サンプル……

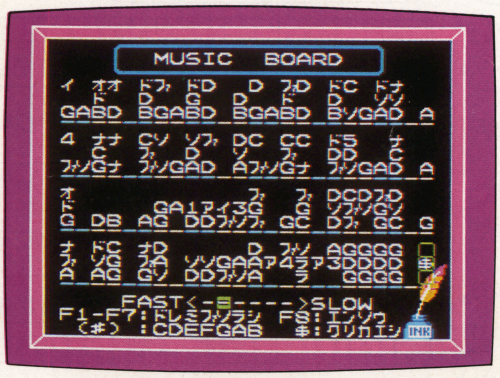
● 下の曲の他にも、いろいろな曲を入力することができます。工夫して楽しんでください。



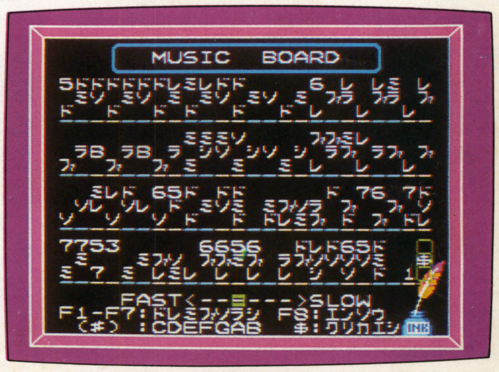
HAPPY BIRTHDAY TO YOU



ジングルベル



ドラえもんの歌



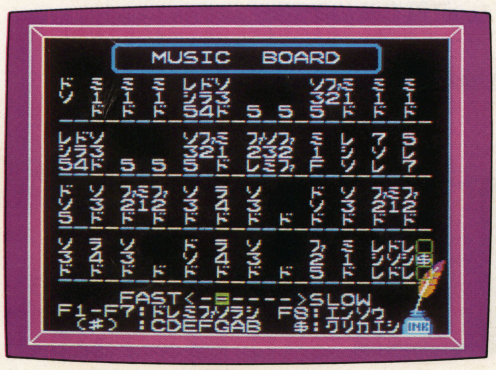
サザエさん



オモチャのチャチャチャ



オリジナル　ロックンロール



ビバルディ　四季

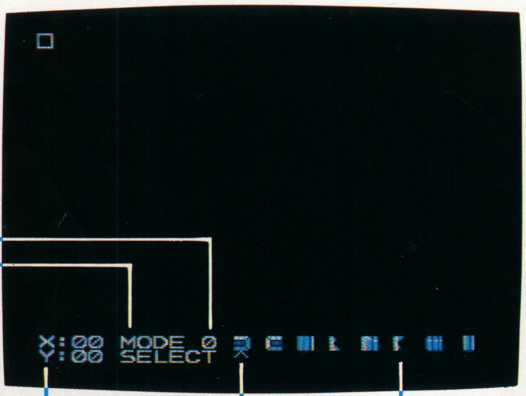


ショパン　幻想即興曲

GAME BASICモード画面

BG　GRAPHICの画面で背景を描き、BASICの画面でゲームのプログラムを作ろう。2つの画面を合わせて、オリジナルゲームが楽しめます。

BG GRAPHICの画面



配色番号

画面にキャラクタを配置する前にRETURNキーを押して、好みの配色を選びます。4種類まで選べます。

実行モード表示

ファンクションメニューで選んだモードが表示されます。

座標値表示

カーソルがどの位置にあるか表示します。(SELECT、COPY、MOVE時のみ)
▲▼◀▶キーでカーソルを動かします。
キャラクタの描ける画面の範囲は、横28マス×縦21行です。
●X(横)は0〜27マス
●Y(縦)は0〜20行

キャラクタマーカー

DELキーを押すと右へ移動します。INSキーを押すと左へ移動します。表示したいキャラクタを選びます。

キャラクタグループ

この中から表示したいキャラクタを選びます。8個のグループを変えるにはCLR HOMEキーを押します。
SHIFTキーを押しながらCLR HOMEキーを押すと、8個のグループが逆方向に変わります。

※すべてのキャラクタは、113ページの「キャラクタテーブル　B」をご覧ください。

ESCキーを押すと、ファンクションメニューが表示されます。
画面を編集したりファイルしたりするためのモードを選びます。



ファンクションメニュー

セレクト SELECT	キャラクタを選びます。Dキーでキャラクタを消します。
コピー COPY ムーブ MOVE	画面内でキャラクタを編集します。
クリア CLEAR	画面に表示してあるキャラクタをすべて消します。
ファイル FILE	画面のデータをカセットテープにSAVE(保存)、カセットからLOAD(呼び出し)します。
キャラクタ CHAR	英数字・記号・カナを入力します。

▲▼キーでメニューを選び、SPACEキーでメニューを実行します。

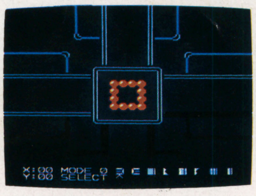
※BG　GRAPHICのファンクションメニューの詳しい使い方は40ページをご覧ください。

BG　GRAPHICでゲームの背景を描くには…

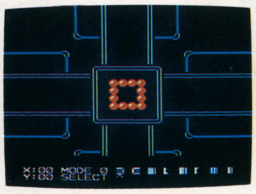
操作手順

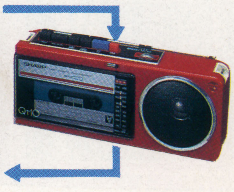
1. BG　GRAPHICの最初のモードはSELECTです。
2. CLR HOMEキー、またはSHIFTキーを押しながらCLR HOMEキーを押して、8個のグループを選びます。
3. INS、DELキーを押して、表示したいキャラクタを選びます。
4. RETURNキーを押して、キャラクタの配色を決めます。
5. ▲▼◀▶キーを押して、表示したい位置を決めます。
6. SPACEキーを押して、キャラクタを画面に描きます。

描画途中の背景画(例)



完成した背景画(例)





画面に描いた背景画をカセットテープへSAVE(保存)することができます。

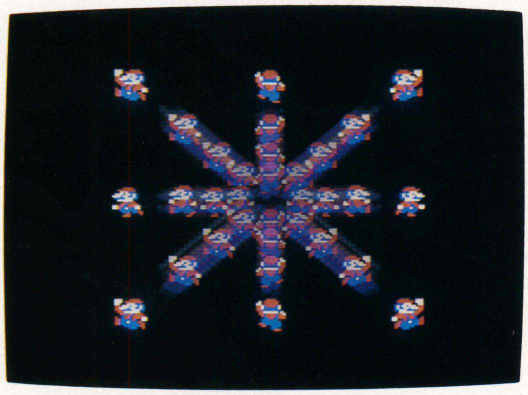
カセットテープからLOAD(呼び出し)して、背景画を画面に表示することができます。

※詳しくは45ページをご覧ください。



GAME BASICモード画面の操作方法

BASICの画面



スピード感のあるゲームにピッタリ! プログラムはすご～く簡単

●キャラクタを動かすのも数行のプログラムでできます。

愉快なアニメキャラクタが同時に動かせる

●BASIC命令でMOVE、DEF SPRITEを使うと、一度に16種類のキャラクタが扱え、迫力ある画面が楽しめます。

アニメキャラクタを背景にかくすことができる

●アニメキャラクタに、表示する優先順位をつければ、背景にかくれたり、現われたりします。

コントローラでアニメキャラクタを操作できる

●コントローラを利用するプログラムをつくって、2人でゲームを楽しむこともできます。

マリオを8方向に動かすには…

```
10 SPRITE ON RETURN
20 CGSET 1,0 RETURN
30 FOR N=0 TO 7:DEF MOVE(N)=SPRITE(0,N+1,3,255,0,0):NEXT
   RETURN
40 MOVE 0,1,2,3,4,5,6,7 RETURN
RUN RETURN
```

マリオを歩かせるのも、わずか数行のプログラムでOK!

※31ページにMOVEコマンドの使い方を記載してありますのでご覧ください。

アニメキャラクタと背景の絵を合成する方法は…

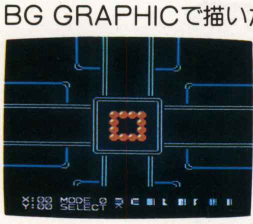
BASICでプログラムしたアニメキャラクタをBG GRAPHICで描いた背景画と合成させるには

●プログラムにVIEW命令を入れる(例 15 VIEW RETURN)

●ダイレクト命令として単にBG GRAPHIC画面をコピー表示させる(VIEW RETURN)

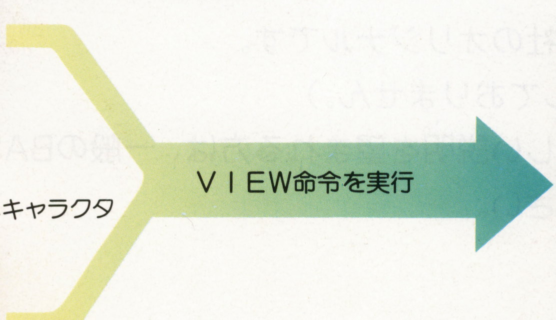
の方法があります。

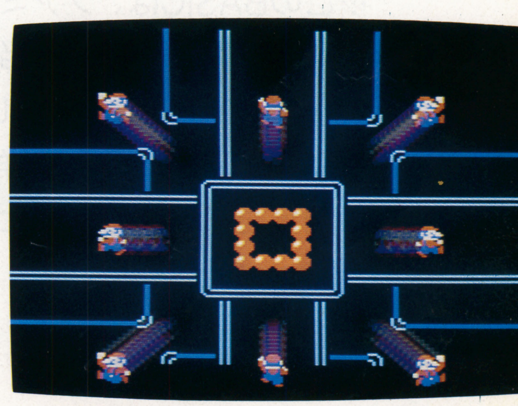
BG GRAPHICで描いた背景画



BASICで作成したアニメキャラクタ







(背景画とアニメキャラクタを合成した画面)

※BASICのプログラミングのしかたは、次のページ以降で詳しく説明してあります。

GAME BASIC

さあ、いよいよオリジナルゲームを作るための「GAME BASIC」の説明に入ります。ファミリーベーシックの特長として、たいへん簡単なプログラムを入力するだけで、すぐにマリオやレディを動かすことができ、楽しく遊べるようになっています。
まず、基本的なプログラムを覚えてください。

ここでは下記の順で説明しています。

1] 入門編

- BASIC
  マリオやレディなどのアニメキャラクタを画面に登場させ、色を変えたり、動かしたりするプログラムの入力の仕方。
- BG GRAPHIC (BACKGROUND GRAPHICの略) 背景や絵の描き方。
- BASICとBG GRAPHIC アニメキャラクタと背景を合成する方法。

※ここまでで基本的なことはすべてマスターできるようになっています。
あとは、英数字・記号を入れ替えて応用するだけで、オリジナルゲームを作ることができます。

2] 文法編

- このファミリーベーシックで使える、すべてのコマンド(命令)を詳しく説明しています。
  入力中にわからないことがあった時は、ここを見てください。

3] ゲームプログラム編

- ゲームの例題を用意しています。参考にしてください。

※プログラム等を入力している途中で、もしコンピュータの状態がわからなくなった時は、19ページの下段をご覧ください。

※このBASICは、ファミリーコンピュータの特長を最大限に活かすように開発された、当社のオリジナルです。
(一般の事務用には適しておりません。)
※基本コマンドのより詳しい説明を望まれる方は、一般のBASIC入門書を参考にしてください。

BASIC 入門編

※このファミリーベーシックには付属品として「キャラクタマップ」が付いています。
これは取扱説明書の裏表紙で紹介しているキャラクタテーブルAと、113ページのキャラクタテーブルB、およびカラーチャートを1枚のマップ（シート）にしたものです。
プログラムを入力する時に、すべてのアニメキャラクタや背景用キャラクタ、そして色の組み合わせが一目でわかる、たいへん便利なマップです。
参考にしてください。

BASIC

※BASICで、画面に表示できるすべてのアニメキャラクタを、うら表紙のキャラクタテーブルAで紹介しています。

■ダイレクトモードについて

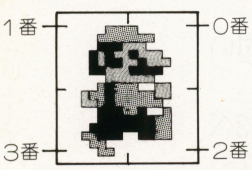
さて、今からコンピュータにBASICの命令を簡単に実行させる方法を紹介します。
キーボードからBASICの命令を入力した後、[RETURN]キーを押すだけです。正しい命令であれば、コンピュータは画面にOK(オーケー)を表示します。正しくなければ、?SN ERROR(シンタックス エラー)などのエラーメッセージを表示します。この場合には、[▲][▼][◀][▶]キーを使い、カーソルを間違った命令の所まで移動し、正しい命令に書き直してから、もう一度[RETURN]キーを押します。または、改めて正しい命令を入力した後、[RETURN]キーを押してください。
このように、キーボードから直接命令を入力して実行させることをダイレクトモード(直接モード)と呼びます。
さあ、キーボードのキーを押して、マリオを画面に出したり、動かしたりしてみましょう。
※これからのプログラム入力方法は、アニメキャラクタの中からマリオを使って説明していきます。

★マリオを画面に表示させる準備

SPRITE ON [RETURN]

を入力します。これでマリオを画面に表示させる準備ができました。

★アニメキャラクタを決めるには…



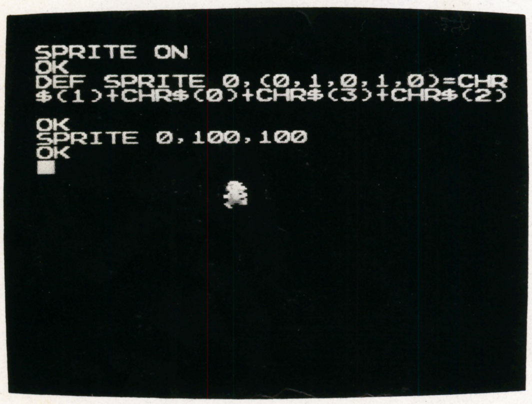
CHR$(0):右上
CHR$(1):左上
CHR$(2):右下
CHR$(3):左下

(マリオ(WALK 1)を左右入れ替えて使っています。)

DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+CHR$(0)+CHR$(3)+CHR$(2) [RETURN]

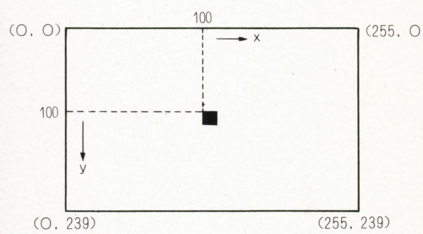
を入力してください。
この命令文は、4つのキャラクタで構成されているスプライト(アニメキャラクタ)を指定する命令です。
これで、マリオはいつでも表示できる状態になりました。

★マリオの表示位置を決めよう



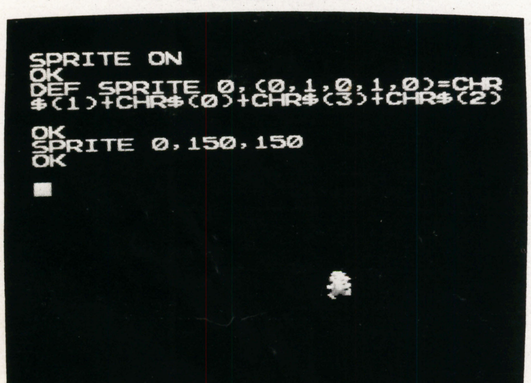
SPRITE 0,100,100 [RETURN]

を入力すると、マリオが画面中央付近に表示されます。



この命令は、マリオを画面の座標(x:100, y:100)の位置に表示しなさいということです。
命令を入力するときは、4ページのキーボードの図を見て、英数字・記号を正しく入力することが大切です。
もし、誤って入力した場合は、[▲][▼][◀][▶]キーを使って、カーソルを誤っている所まで移し、正しい命令を入力してください。

★マリオの表示位置を自由に変えてみる



SPRITE 0,150,150 [RETURN]

を入力すると、先ほどより少し右下にマリオが表示されます。

SPRITE命令の100, 100を150, 150に修正する方法

▽[▲]キーを押して、カーソルをSPRITEの行に移します。
 SPRITE 0, 100, 100
▽[▶]キーを押して、修正したい箇所にカーソルを移します。
 SPRITE 0, 100, 100
▽[5]キーを押すと、0が5にかわります。
 SPRITE 0, 150, 100
▽続いて、[▶]キーを押して、もう1箇所の所にカーソルを移します。
 SPRITE 0, 150, 100
▽[5]キーを押すと、0が5にかわります。
 SPRITE 0, 150, 150
▽ここで、[RETURN]キーを押すと、命令が書き換えられます。

同じような操作を行なって数字を変えると、マリオを好みの位置に表示することができます。

★マリオの色を変える

CGSET 1,1 [RETURN]

を入力すると、これまで表示されていた色のマリオとは違い、赤い帽子をかぶったマリオになりましたね。

■プログラムモードについて

ここまでは、コンピュータに直接命令を与える方法でコンピュータに仕事をさせました。これをダイレクトモードといいます。この場合、1つの命令を入力すると、1つの仕事しか行なわないので、何回も同じ仕事をさせるには、その度に、命令を入力してやらなければなりませんでした。仕事の順番を命令で、コンピュータのメモリの中に記憶させておけば、手間をかけずに同じ仕事を何回でも行なわせることができます。
このように、仕事の順序をメモリの中に記憶させ、実行させることを、プログラムモードと呼びます。

★プログラムモードとダイレクトモードのちがいは…

```
10 SPRITE ON
20 DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+CHR$(0)+CHR$(3)+CHR$(2)
30 SPRITE 0,100,100
```

(10〜30の各行全体：命令文／行頭の数字：行番号／SPRITE 以下：命令)

上の図は、マリオを表示させるための手順の一例を示したものです。
プログラムモードでは、命令を実行させる順番に番号をつけて、メモリに記憶します。そうすれば、何度でも記憶したプログラムを呼び出して、簡単に実行させることができます。
つまり、命令に実行させる行番号をつけて、メモリに記憶させ、実行させることをプログラムモードと呼びます。行番号のついた命令の集まりがプログラム（命令文）です。
コンピュータは、小さい行番号の順から正確にプログラムを実行します。
上の図に示してあるように、行番号（整数）は飛び飛びの番号をつけておくと、プログラムを追加したり、変更したりするときに便利です。また、プログラムの中では、同じ番号は使えません。通常、プログラムを作る場合には、10ずつ増した行番号をつけておくことをおすすめします。

★プログラミングする準備をしよう。

プログラミングする前に、

CLS RETURN ……（バックグラウンド面に表示された英数字、カナ、記号、または背景を消してカーソルをホームポジション（画面左上）に戻す命令です）

を入力し、画面に表示されている命令を消します。
また、

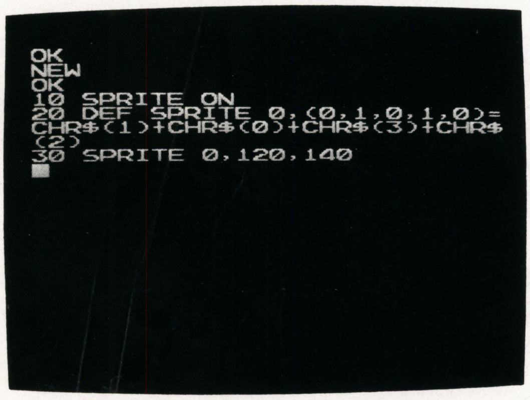
NEW RETURN ……（記憶されているプログラムをすべてメモリから消去する命令です）

を入力して、コンピュータに入力されていたプログラムをメモリから消しておきます。前の不要なプログラムを消す場合は、この命令を入力してください。

CTR + D ……（CTRキーを押しながらDキーを押します）　※19ページをご覧ください。

スプライト面を消したり、色をあらかじめ与えられた配色に戻します。（マリオが出ている場合は、消えます）

★プログラムモードでマリオを呼びだそう




```
10 SPRITE ON RETURN
20 DEF SPRITE 0,(0,1,0,1,0)=
   CHR$(1)+CHR$(0)+CHR$(3)
   +CHR$(2) RETURN
30 SPRITE 0,120,140 RETURN
```

を入力してください。

上で入力したプログラムを行番号順に説明します。

10 SPRITE ON（オン）
アニメキャラクタの表示ができるようになります。一度使用すると、SPRITE OFF（オフ）命令を出すまで有効です。

20 DEF SPRITE
＝より右にある文字式から、アニメキャラクタを構成し、定義します。

DEF SPRITE Ø,(Ø,1,Ø,1,Ø)＝CHR$(　)…………

- アニメキャラクタ番号
- カラー
- 組み合わせ型
- 表示優先度
- アニメキャラクタの反転指示それぞれX、Y方向（このプログラムではX方向の反転指示）

※CHR$(　)内の数字を入れ替えると、他のアニメキャラクタに変更することができます。

30 SPRITE Ø, 120, 140
アニメキャラクタ番号Øが、座標X：120，Y：140の位置に表示されます。

★プログラムを実行させるには…



RUN [RETURN]
を入力してください。([F8]キーを押すだけでも命令が実行されます)
画面にマリオが表示されましたね。

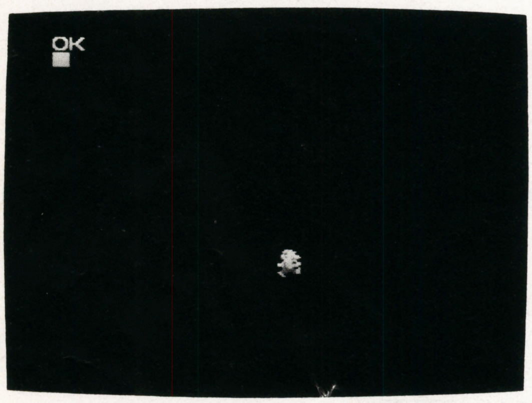
●ファンクションキーについて…

[F1]〜[F8]までのファンクションキーには、プログラムを作成するときに便利な命令があらかじめ定義されています。

[F1]…LOAD [RETURN]	[F5]…SPRITE
[F2]…PRINT	[F6]…CONT [RETURN]
[F3]…GOTO	[F7]…LIST [RETURN]
[F4]…CHR$(	[F8]…RUN [RETURN]

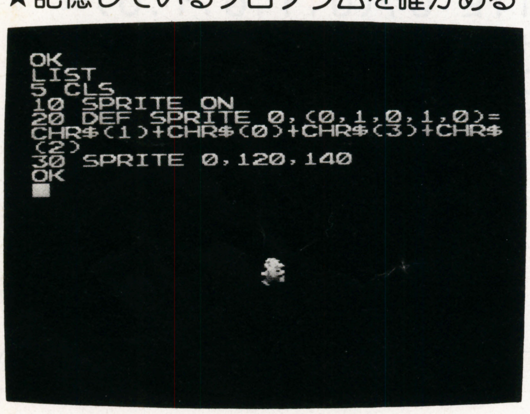
※[F1]〜[F8]の１つのキーを押すだけで、各ファンクションキーに定義された内容が実行されます。
(ただし、[F1]、[F6]、[F7]、[F8]については、表示、実行されます)
※それぞれの命令を覚えておけば、プログラムの入力がスムーズに進みます。

★画面にマリオだけを残すには…



17ページの「プログラムモードでマリオを呼びだそう」のプログラムを利用して、
5 CLS [RETURN]
この命令文を入力してください。
RUN [RETURN]
を入力してください。
先ほど入力した命令文は消え、マリオだけが画面に残りましたね。

★記憶しているプログラムを確かめる



LIST [RETURN] …(記憶しているプログラムを画面に表示させる命令です)
を入力してください。
画面には、左図のようなプログラムが表示されます。
コンピュータは、追加入力した
5 CLS
のプログラムをきちんと整理して記憶しています。
このように、プログラムは、行番号をつけて命令を入力してやれば、あとからでもプログラムを追加することができます。

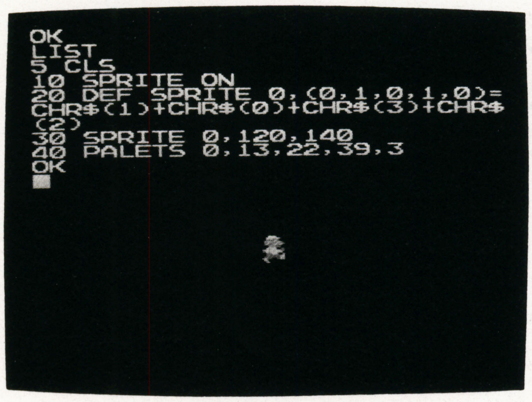
●LIST(リスト)について…

LISTとは、記憶しているプログラムを画面に表示する命令です。

LIST m－n
(m：表示するプログラムの開始行番号)
(n：表示するプログラムの終了行番号)

LIST [RETURN]	記憶しているプログラムをすべて表示します。
LIST 30 [RETURN]	行番号30のプログラムを表示します。
LIST 20－ [RETURN]	行番号20以降のプログラムを表示します。
LIST 20－30 [RETURN]	行番号20から30までのプログラムを表示します。
LIST －30 [RETURN]	プログラムの先頭から行番号30までを表示します。

★マリオの色を変える



プログラム1

18ページの「画面にマリオだけを残すには…」のプログラムを利用して、

40 CGSET 1,1 RETURN

を入力してください。

RUN RETURN

を入力すると、先ほどのマリオの色が変化します。

40 PALETS 0,13,22,39,3 RETURN

を入力してください。

RUN RETURN

を入力すると、少しちがった色のマリオが表示されます。

LIST RETURN

を入力して、先ほど入力したPALETS命令が記憶されていることを確認してください。行番号40のPALETSを入力すると、前のプログラム（行番号40のCGSET 1,1）が書き換えられます。

▲ ▼ ◀ ▶ キーを押して、カーソルをPALETS命令の行に移してください。マリオや背景の色を変えてみるため、PALETS命令を、次のように変えます。

40 PALETS 0,17,22,39,3 RETURN

を入力した後、

RUN RETURN

を入力してください。背景に色がつきましたね。

ここでもとのマリオの配色にもどしておきましょう。

40 PALETS 0,13,54,22,2 RETURN

RUN RETURN

を入力してください。

●CGSET（シージーセット）について…

CGSETとは、あらかじめ用意された色の組み合わせのグループの中から好きなグループの色を指定する命令です。

CGSET m,n
- n: アニメキャラクタ（スプライト）のパレットコード（0～2番でグループを指定します）
- m: バックグラウンド面のパレットコード（0～1番でグループ指定します）

●PALET（パレット）について…

PALETとは、52色の中から選んでアニメキャラクタや背景の絵柄を表示する命令です。PALETには2種類あります。

PALETS（パレットエス）……アニメキャラクタ（スプライト）用
PALETB（パレットビー）……背景の絵柄用

PALETS 0, 17, 22, 39, 3
- 17: バックドロップ面（背景）の全体の色を指示します。バックドロップ面の色は、あとで実行指定した命令が有効となります。
- 17, 22, 39, 3: 52色のカラーコードを指示します。数字の範囲は0～60です。

※113ページのカラーチャートおよび73ページの、52色の色コードをご覧ください。

●CTR（コントロール）+D（デー）について…

CTR+D とは CTRキーを押しながらDキーを押すことを意味します。
これを実行すると、CGEN（シージェン） 2、SPRITE OFFの命令を実行します。
カラーパレットは、バックグラウンド用、スプライト用共にバックグラウンド用のパレットコードの配色となります。
CTR+A（オートインサート機能）を解除します。
ただし、変数の値はクリアされず、そのままです。
この機能を利用すれば、プログラムを変更する場合、画面に表示されているスプライト（アニメキャラクタ）を消したり、文字の色を白色に戻すことができますので、画面が見やすくなります。

※CGEN 71ページをご覧ください。
※SPRITE OFF 27ページをご覧ください。

※104ページのコントロールコード表をご覧ください。

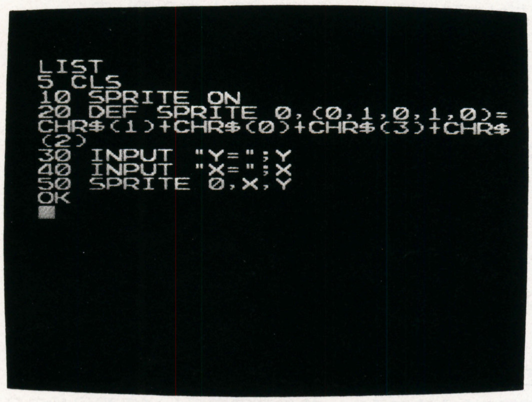
●もし、コンピュータの状態が、わからなくなったときは…

BASICでプログラムを実行しているとき、カーソルが見えなくなったり、コンピュータの状態がどうなっているかがわからなくなった場合、次の操作を行なってください。

①STOPキーを押す
②CTR+Dキーを押す
③SHIFT+CLR HOMEキーを押す

この操作で、何も表示されていない画面左上にカーソルが点滅している状態に戻ります。もし、これらの操作を行なっても、画面がこの状態に戻らない場合は、コンピュータがコントロール不可能な状態になっています（この状態を、コンピュータが暴走したと呼びます）。ファミリーコンピュータのリセットスイッチを押して、スタート画面から操作をやり直してください。
POKE命令やCALL命令を使ったプログラム中で誤りがあった場合に、発生することがありますので、ご注意ください。

★変数を使ってマリオを動かそう

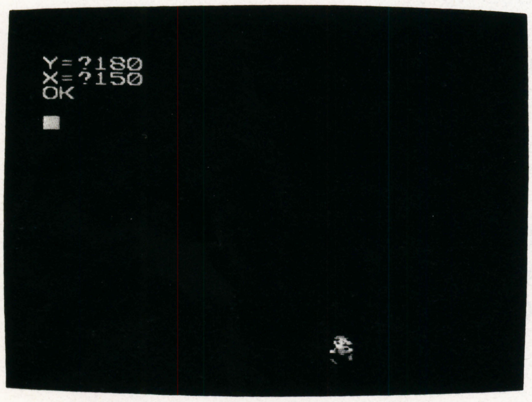


プログラム2

19ページのプログラム1を使って

30 INPUT"Y=";Y RETURN
40 INPUT"X=";X RETURN
50 SPRITE 0,X,Y RETURN

を入力してください。

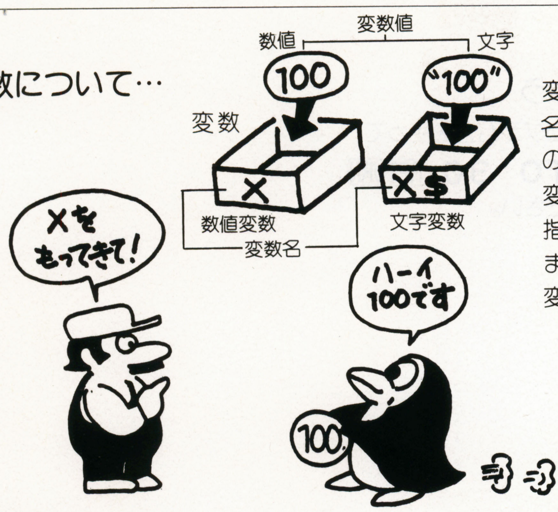


RUN RETURN
を入力すると、
Y=? ■
が表示されますので、そのまま、その行に
180 RETURN
を入力してください。すぐその下に
X=? ■
が表示されますので、そのまま、その行に
150 RETURN
を入力してください。
座標(150, 180)の位置にマリオが表示されます。
何回も実行させたい場合は、
RUN RETURN を入力してください。
命令で使用できる数字には制限があります。
X:Ø~255, Y:Ø~255(ただし、表示できるのはX:Ø~240, Y:5~220)の範囲です。
※テレビ受像機によって、表示できる範囲が変わります。

● INPUTについて…
INPUTとは、キーボードから数値や文字のデータが入力されるのを待ち、入力すると指定された変数にそのデータを代入する命令です。
上のプログラムで使ったINPUT命令には、"(ダブルクォーテーション)マークが使用されています。
INPUT "Y=";Y
"で囲まれたY=の記号はそのまま画面に表示されます。
※60ページをご覧ください。

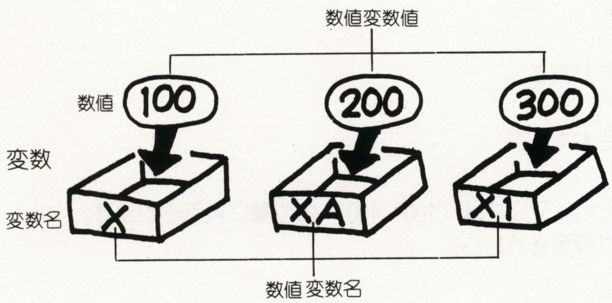
●PRINTについて…
PRINTとは、画面に文字や計算の結果を表示する命令です。
PRINT
1行分の空白をつくります。
A$="MARIO SAMPLE V. 100.0":B$="ノ プログラム"
PRINT A$+B$
文字変数A$とB$を1行に続けて表示します。(MARIO SAMPLE V.100.0ノ プログラム と表示します)
PRINT 100+100 RETURN
を入力すると、1ØØ+1ØØ を計算して答200を表示します。(ダイレクトコマンド)
※59ページをご覧ください。

●変数について…



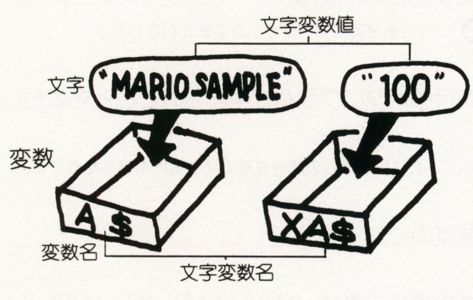
変数とは、数値・文字などのデータを入れる箱です。この箱には、名前をつけて区別しておきます。この場合、箱の名前を**変数名**、箱の中に入れる値を**変数値**と呼びます。
変数名をつけて、その箱の中へ変数値を入力しておけば、変数名を指定するだけで、その中に入っている変数値を取り出すことができます。
変数には、**数値変数**と**文字変数**があります。

数値変数について…



- 数値変数の変数名は、XやYなどの記号で名前をつけます。変数名は255文字までの記号が扱えますが、コンピュータは最初の2文字だけで変数名を区別します。（長い変数名をつけても、2文字以降はコンピュータは判別しません）
  また最初の1文字目の記号は、必ず英文字を使います。ただし、英文字であってもBASICで使う命令は使えません（LIST、RUN、PRINTなど）
- 変数値には、100や200などで表わされる整数を使います。
  例えば、
  X=100，XA=200，X1=300（数値変数名：X、XA、X1　数値変数値：100、200、300）
  なお、数値変数にデータを入れる前は、変数値には0の数値が入っています。

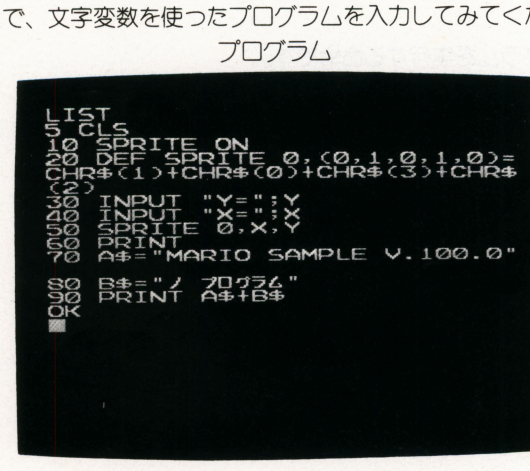
文字変数について…



- 文字変数の変数名は、XやYの記号の後に$（ダラーマーク）を付けて数値変数と区別します。変数名は数値変数と同様に255文字までの記号が扱えますが、コンピュータは最初の2文字だけで、変数名を区別します。
  また、最初の1文字目の記号は、必ず英文字を使います。
- 変数値は、"（ダブルクォーテーション）でくくった型で入力します。この変数値は、数値変数の場合とは異なり、数字であってもすべて文字として扱われます。
  例えば、
  A$="MARIO SAMPLE"　XA$="100"（文字変数名：A$、XA$　文字変数値：MARIO SAMPLE、100）
  なお、変数値にデータが与えられるまでは、箱の中には何も入っていません。

ここで、文字変数を使ったプログラムを入力してみてください。

プログラム



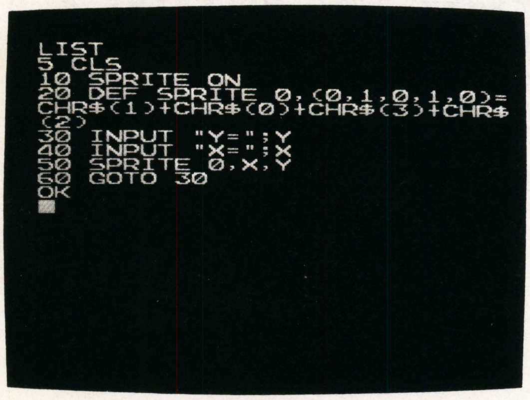
実行画面の例



このプログラムで使用されている文字変数名は、A$、B$です。
文字変数値は、"MARIO SAMPLE V.100.0"と"ノ プログラム"です。

◆繰り返しのテクニックを覚える

★GOTO文でマリオの位置をつぎつぎに変えてみよう

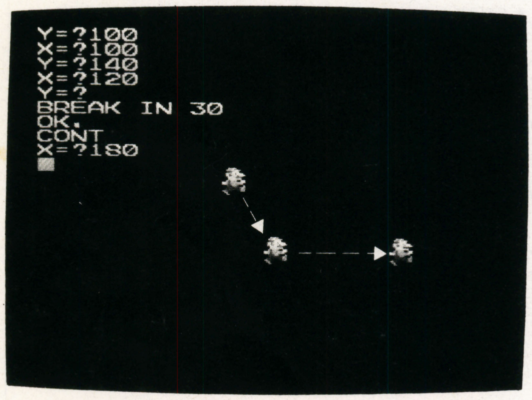


プログラム３

20ページのプログラム２を使って、

60 GOTO 30 RETURN

を入力してください。



RUN RETURN

を入力してください。

Y=?　が表示されたら、

100 RETURN

X=?　が表示されたら、

100 RETURN

を入力してください。マリオは（100，100）の位置に表示されます。

再度、Y=?　が表示されたら、

140 RETURN

X=?　が表示されたら、

120 RETURN

を入力してください。マリオは（100，100）から（120，140）に表示位置を変えます。

再び、Y=?　が表示されたら、

プログラムの実行を停止させる場合は、STOPキーを押してください。

BREAK IN 30……（行番号30でプログラムが中断されました。）

が表示されます。

これは、行番号30の命令を実行中、プログラムが中断されたことを表わしています。

CONT RETURN……（中断していたプログラムの続きをそのまま実行させる命令です。）

を入力してください。

再度、X=?　が表示されますので、

180 RETURN

を入力してください。マリオの表示位置は、（120，140）から（180，140）へ変わります。

●GOTO（ゴーツー）について…

GOTOとは、決められた行番号へ強制的に飛び、何回でもプログラムを実行する命令です。

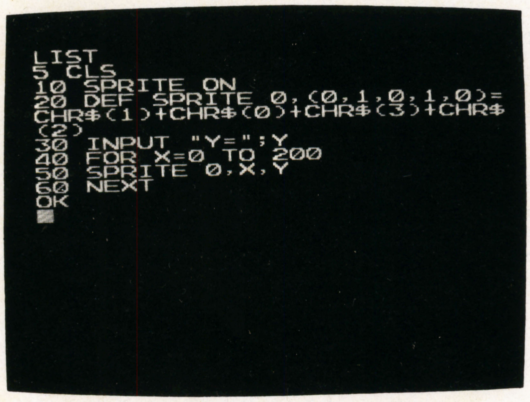
何回もくり返す

```
→30  INPUT "Y=";Y
 40  INPUT "X=";X
 50  SPRITE 0,X,Y
 60  GOTO 30
```

行番号50で、マリオを表示したら、強制的に行番号30へ飛び、実行します。

※63ページをご覧ください。

★FOR　NEXT文でマリオを横方向に移動させよう



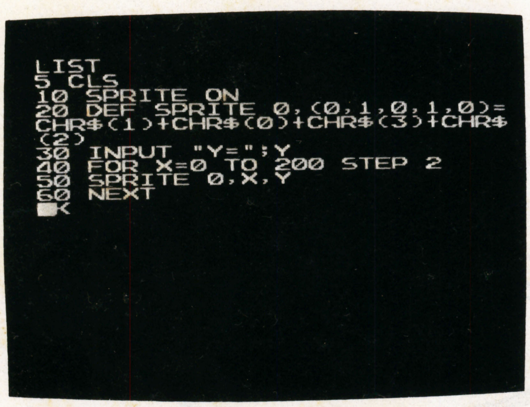
22ページのプログラム3を使って、

40　FOR　X=0　TO　200 RETURN
60　NEXT RETURN

を入力してください。
※FOR NEXTは次のページの「多重ループについて」で説明してあります。



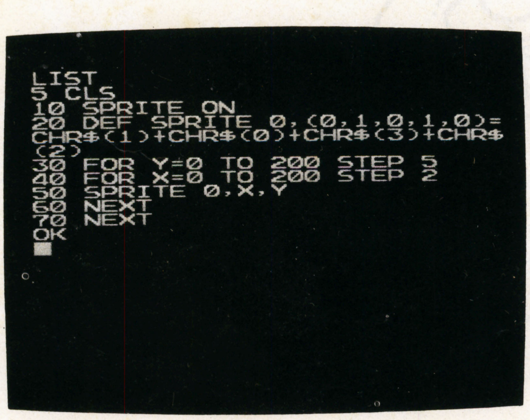
RUN RETURN
を入力してください。
Y=?　が表示されたら、
100 RETURN
を入力してください。マリオが左から右へ動きます。



LIST RETURN
を入力してください。
▲▶キーを押して、カーソルを行番号40に移し、命令を書きかえてください。

40　FOR　X=0　TO　200　STEP　2 RETURN

次に、▼キーを押して、カーソルを行番号60の次の行に移してください。
RUN RETURN
を入力すると、Y=?　が表示されますので、
100 RETURN
を入力してください。先ほどとは違って、速くマリオが左から右へ動きますね。

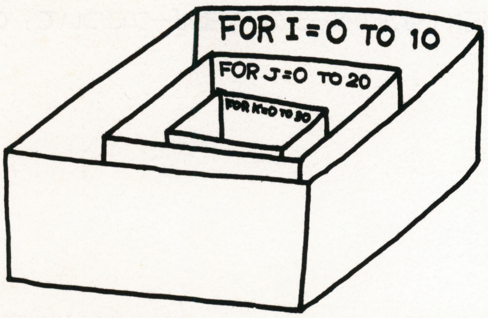


プログラム4

30　FOR　Y=0　TO　200　STEP　5 RETURN
70　NEXT RETURN
を入力してください。
RUN RETURN
を入力すると、マリオは画面左上から画面右下の方へ移動します。

●多重ループについて…

FOR～NEXTは、決まった回数だけプログラムを繰り返し実行します。つまり、FOR～NEXTの繰り返しは、1つのループを構成しているのです。このループの中に、もう1つのFOR～NEXTを使用することができます。このように、FOR～NEXTのループの中に何個ものFOR～NEXTを構成することを多重ループと呼びます。




```
①FOR I=0 TO 10
 ②FOR J=0 TO 20
  ③FOR K=0 TO 30
      :
   NEXT
      :
  NEXT
      :
 NEXT
```
ループ

多重ループは、一番外側にあるFOR～NEXTから順に実行されます。
ただし、1つのループを横ぎるようなループは、プログラムとしては指示できません。

```
FOR I=0 TO 100
 FOR J=0 TO 30
NEXT
 NEXT
```

また、GOTO命令を使って、FOR～NEXTの間へ割り込ませて使うことはできません。

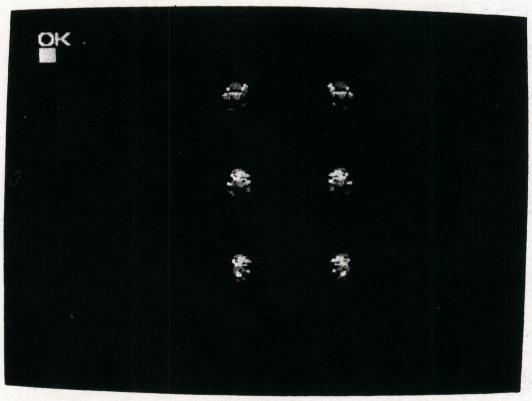
※65ページをご覧ください。(FOR～TO～STEP NEXT)



★いろいろなポーズのマリオを表示させよう

さあ、いろいろなポーズをしたマリオを、画面に表示させてみましょう。
23ページのプログラム4を使って、次のプログラムを入力してください。

```
15 CGSET 1,0 RETURN
21 DEF SPRITE 1,(0,1,0,0,0)=CHR$(0)+CHR$(1)+CHR$(2)+
   CHR$(3) RETURN
22 DEF SPRITE 2,(0,1,0,1,0)=CHR$(5)+CHR$(4)+CHR$(7)+
   CHR$(6) RETURN
23 DEF SPRITE 3,(0,1,0,0,0)=CHR$(4)+CHR$(5)+CHR$(6)+
   CHR$(7) RETURN
24 DEF SPRITE 4,(0,1,0,0,0)=CHR$(20)+CHR$(21)+CHR$(22)
   +CHR$(23) RETURN
25 DEF SPRITE 5,(0,1,0,1,0)=CHR$(21)+CHR$(20)+CHR$(23)
   +CHR$(22) RETURN
30 SPRITE 0,100,100 RETURN
40 SPRITE 1,150,100 RETURN
50 SPRITE 2,100,150 RETURN
60 SPRITE 3,150,150 RETURN
70 SPRITE 4,100,50 RETURN
80 SPRITE 5,150,50 RETURN
```



RUN RETURN
を入力してください。
画面には、6種類のポーズのマリオが表示されます。
※いろいろなポーズのマリオについては、裏表紙の「キャラクタテーブルA」をご覧ください。





ここまで読めば、BASICプログラムとは何か、どうすれば動くかがわかったと思います。つまり、BASICについての基本的な事がらを学んだことになります。
もう、あなたはBASICのプログラマー。簡単なプログラムなら、すぐに作成できるでしょう。チャレンジしてください。
この後、26ページ~30ページでは、BASICの基本を発展させた、コントローラでアニメキャラクタを自由に動かすプログラムについて説明しています。

✲アニメキャラクタを表示する方法について

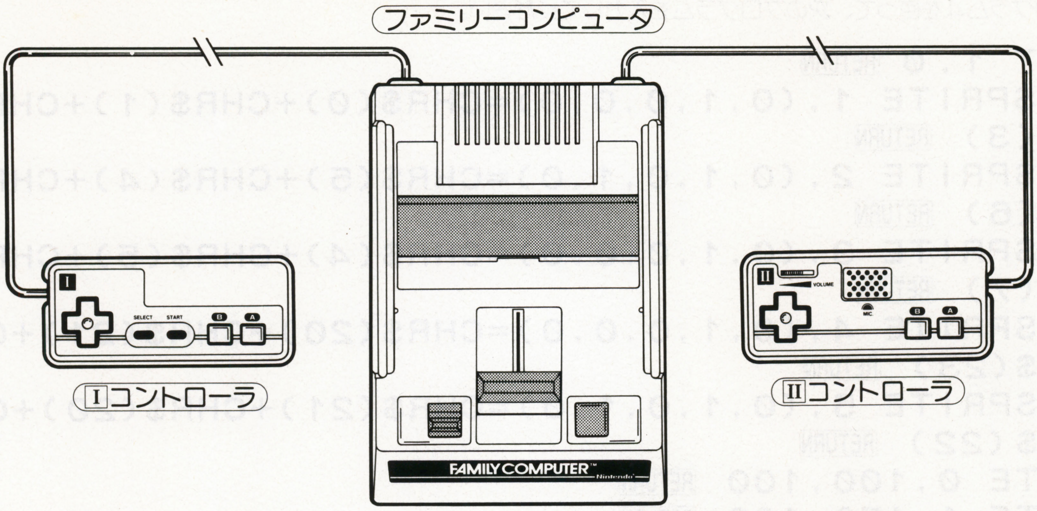
●DEF MOVE

これまでは、DEF SPRITE(デファイン スプライト)命令で説明してきましたが、もう1つの命令を紹介します。
それは、DEF MOVE(デファイン ムーブ)命令と呼ぶものです。

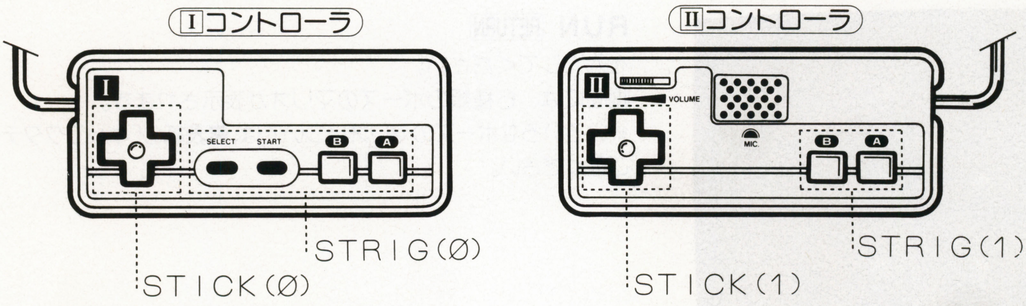
- これは、キャラクタテーブルAにある16種類のアニメキャラクタを簡単に動かせるスマートな命令です。
- この16種類の指定をDEF MOVE命令で行なえば、自動的にアニメキャラクタが合成されて動作を表示できるようになります。
- その他、方向,速さ,移動距離,カラー,表示優先度が指定できます。
- この命令の使い方は「MOVEコマンドについて」の項で説明しています。※詳しくは32ページをご覧ください。
- 初めに説明したDEF SPRITE命令を使って自由な組み合わせのアニメキャラクタを作り、さらに、簡単なDEF MOVE命令を使うと、同時に、全部で16個の異なったアニメキャラクタを表示できることになります。

◘コントローラで動かそう

コントローラを使って遊べるようなゲームのプログラムを作れば、2人でゲームをして遊ぶこともでき、さらにゲームの楽しさが拡がります。



コントローラを操作する命令には、STICK（スティック）とSTRIG（エス トリガー）の2つがあります。STICKは、コントローラの✣ボタンのどの部分が押されたか（押された方向）をコンピュータに数値で与える命令です。STRIGは、コントローラの4種類のボタン（スタート・セレクト・A・B）のどれが押されたかをコンピュータに数値で与える命令です。
Iコントローラを使用する場合は、STICK（Ø）やSTRIG（Ø）の命令、IIコントローラを使用する場合は、STICK（1）やSTRIG（1）の命令を使います。

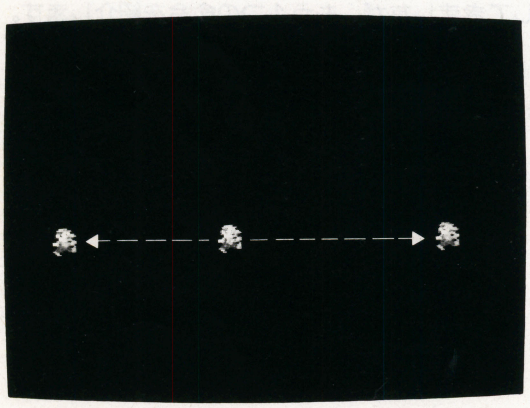


★コントローラでマリオを左右に動かそう

SPRITE OFF [RETURN] を入力して、表示しているマリオを消しておきます。25ページの「いろいろなポーズのマリオを表示させよう」のプログラムを利用して、次のプログラムを入力してください。

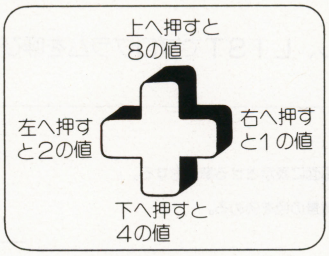
```
26  X=100 [RETURN]
30  S=STICK(Ø) [RETURN]
40  IF  S>2  THEN  100 [RETURN]
50  IF  S=2  THEN  X=X-1 [RETURN]
60  IF  S=1  THEN  X=X+1 [RETURN]
70  IF  X>250  THEN  X=X-240 [RETURN]
80  IF  X<5  THEN  X=X+245 [RETURN]
90  SPRITE 0,X,150 [RETURN]
100  GOTO 30 [RETURN]
```

RUN [RETURN]
を入力してください。
Iコントローラの✣ボタンを押す（左側や右側）と、マリオが左や右に動きます。
[STOP]キーを押すと、カーソルが出てきます。



●STICKについて…

STICKとは、コントローラの✚ボタンがどの方向に押されたかをコンピュータに数値で与える命令です。



コントローラの ✚ボタン

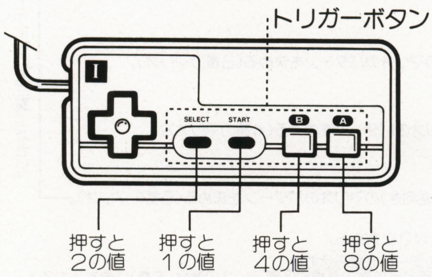
S=STICK(0)は、……Ⅰコントローラ（STICK(1)はⅡコントローラ）

✚ボタンの右側が押されると、Sには1の値が入る。
✚ボタンの左側が押されると、Sには2の値が入る。
✚ボタンの下側が押されると、Sには4の値が入る。
✚ボタンの上側が押されると、Sには8の値が入る。
✚ボタンが押されていないとき、Sには 0の値が入る。

STICK命令を利用し、コントローラの✚ボタンを押すことによって、アニメキャラクタを上下左右に自由に動かすことができます。

※86ページをご覧ください。

●STRIGについて…



STRIGとは、コントローラの4種類のトリガーボタン（スタート・セレクト・A・B）のうち、どのボタンが押されているかをコンピュータに数値で与える命令です。

STRIG(0)…Ⅰコントローラ（STRIG(1)はⅡコントローラ）

セレクト ボタンが押されていると、2の値が入る。
スタート ボタンが押されていると、1の値が入る。
B ボタンが押されていると、4の値が入る。
A ボタンが押されていると、8の値が入る。
何もボタンが押されていないと、0の値が入る。

STRIG命令、つまりトリガーボタンの操作によって、マリオの動きを止めたり、ゲームを終るときなどに使用できます。

※86ページをご覧ください。

●IF~THENについて…

IF文とは、IFとTHENの間に書かれた条件文を判断し、その結果によって次に実行する命令（命令文）を指定します。

```
40 IF [S>2] THEN [100]
50 IF [S=2] THEN [X=X-1]
       条件文        命令文
```

つまり、IF文はコンピュータに自分で判断させ、実行させる働きがあります。

コントローラでマリオを左右に動かすプログラムの中で使用されているIF文について説明します。
行番号40は、もし変数Sが2の値より大きい（✚ボタンの上下部が押されていたとき）ならば、THENの後ろに書かれた命令を実行し、行番号100へ飛び、そこからプログラムを実行します。条件に合わない（Sが2の値より小さい、あるいは等しい）場合には、THEN以降の命令は、実行せずに、次の行番号50を実行します。
行番号50は、もし変数Sが2の値（✚ボタンの左部が押されたとき）ならば、X=X−1でXの値から1を引いて新しくXの値を決めます。(左の方向にマリオを動かすためにXの座標値を1だけ引く)

※64ページをご覧ください。

●SPRITE OFFについて…

SPRITE OFFとは、スプライト面に表示しているすべてのアニメキャラクタを消す命令です。つまり、スプライト面をバックグラウンド面への重ね合わせ表示をやめます。
また、スプライト面上に表示されている特定の番号のアニメキャラクタのみを消す場合には、

SPRITE n……(nの番号で指定されたアニメキャラクタが消えます)

を入力します。
このように、アニメキャラクタを画面から消すには、2通りの方法があります。

※89ページをご覧ください。

★コントローラでマリオを歩かせよう

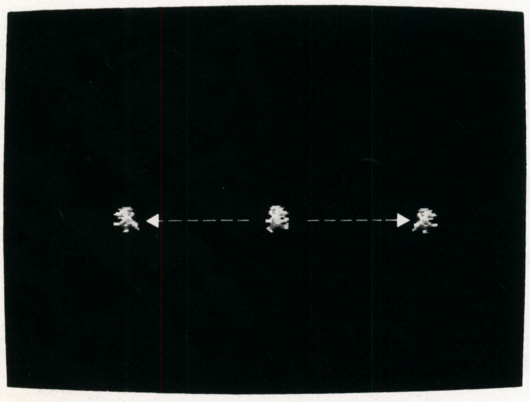
マリオを右や左に歩いているように見せるには、次のプログラムを入力してください。
26ページの「コントローラでマリオを左右に動かそう」のプログラムを中断して、下のプログラム5のうち、行番号 **30 ~ 2020** を入力してください。

※画面にカーソルやプログラムが表示されていない場合は、STOPキーを押してカーソルを出し、LISTでプログラムを呼び出してください。

```
5 CLS
10 SPRITE ON
15 CGSET 1,0
20 DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)
   +CHR$(0)+CHR$(3)+CHR$(2)
21 DEF SPRITE 1,(0,1,0,0,0)=CHR$(0)
   +CHR$(1)+CHR$(2)+CHR$(3)
22 DEF SPRITE 2,(0,1,0,1,0)=CHR$(5)
   +CHR$(4)+CHR$(7)+CHR$(6)
23 DEF SPRITE 3,(0,1,0,0,0)=CHR$(4)
   +CHR$(5)+CHR$(6)+CHR$(7)
24 DEF SPRITE 4,(0,1,0,0,0)=CHR$(20)
   +CHR$(21)+CHR$(22)+CHR$(23)
25 DEF SPRITE 5,(0,1,0,1,0)=CHR$(21)
   +CHR$(20)+CHR$(23)+CHR$(22)
26 X=100
30 READ X,Y,A,B,C,D,E,F
40 DATA 120,140,1,3,0,2,4,5
50 SPRITE A,X,Y
60 S=STICK(0):IF S=0 THEN 60
70 IF S>2 THEN 60
80 X=X+2:IF S=2 THEN X=X-4
90 IF S=1 THEN 1000
100 IF S=2 THEN 2000
1000 PAUSE 5:SPRITE C,X,Y:SWAP C,D
1010 SPRITE A:SPRITE B:SPRITE
     C:SPRITE E:SPRITE F
1020 GOTO 60
2000 PAUSE 5:SPRITE A,X,Y:SWAP A,B
2010 SPRITE A:SPRITE C:SPRITE
     D:SPRITE E:SPRITE F
2020 GOTO 60
```

- 5 ……画面の表示を消す。
- 10 ……アニメキャラクタを画面に表示させる準備をする。
- 15 ……アニメキャラクタや背景の色を決める。
- 20 ……WALK1（右向き）のマリオのパターンを決める（0番のマリオ）。
- 21 ……WALK1のマリオのパターンを決める（1番のマリオ）。
- 22 ……WALK2（右向き）のマリオのパターンを決める（2番のマリオ）。
- 23 ……WALK2のマリオのパターンを決める（3番のマリオ）。
- 24 ……はしごのマリオのパターンを決める（4番のマリオ）。
- 25 ……はしご（左右逆向き）のマリオのパターンを決める（5番のマリオ）。
- （20～25）裏表紙の「キャラクタテーブルA」をご覧ください。
- 26 ……Xの座標を100に決める。
- 30 ……X～Fまでのデータを読み込む。（マリオを表示する位置の座標や各種マリオの番号（変数）に値を入れる）
- 40 ……READ命令で読み込まれるデータ。
- 50 ……1番のマリオを座標（X：120，Y：140）の位置に表示。
- 60 ……✚ボタンが押されているかどうかを調べる。もし押されていなければ（S=0）、くり返し行番号60を実行する。
- 70 ……✚ボタンの上下方向（Y方向）が押されていると行番号60へ飛ぶ。
- 80 ……右方向のときはXの値に2を加える。左方向のときはXの値から4を引く。
- （60～80）Xの座標処理
- 90 ……右方向のときは行番号1000へ飛ぶ。
- 100 ……左方向のときは行番号2000へ飛ぶ。
- （90～100）条件判断
- 1000～1020 0番のマリオを表示する。✚ボタンの右方向が押されていると、0番と2番のマリオを入れ替える。他のマリオ（1，3，0，4，5番）は消す。
- 2000～2020 1番のマリオを表示する。✚ボタンの左方向が押されていると、1番と3番のマリオを入れ替える。他のマリオ（1，0，2，4，5番）は消す。

プログラム5



RUN RETURN

を入力してください。

Ⅰコントローラの✚ボタンを押す（左側や右側）と、マリオはスムーズに、いかにも歩いているように左右へ動きます。

※マリオを画面の左右にはみ出さないようにしてください。はみ出した場合にはエラーとなりマリオはストップしてしまいます。
再度、動かすには **RUN** RETURN を入力してください。

●SWAP（スワップ）について…

SWAPとは、変数の値を互いに交換する命令です。
交換する変数は、数値変数どうし、文字変数どうしでなければエラーとなります。
SWAP　A，A$　や　SWAP　A$，B　のような使い方はできません。

SWAP　A，B………（数値変数Aと数値変数Bの値を交換する。Bのもっていた値がAになり、Aのもっていた値がBとなる）

SWAP　A$，B$…（文字変数A$と文字変数B$の値を交換する）

※67ページをご覧ください。

●マルチステートメントについて…

プログラム5の中で、1つの行番号の中に、：（コロン）を使い2つ以上の命令が書かれているものをマルチステートメントと呼びます。1つの行番号の中には、コロンを含み256文字の命令が書けます。

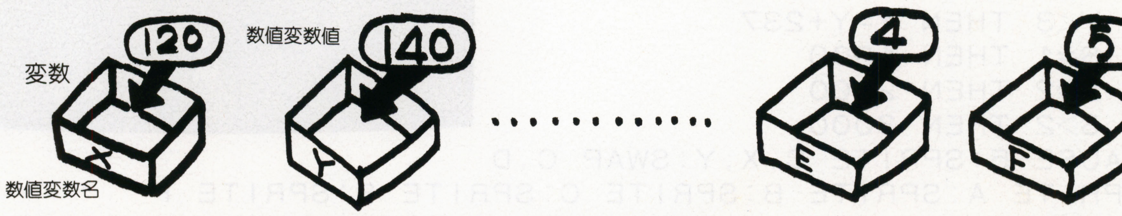
●READ（リード）～DATA（データ）について…

READとは、DATAで用意されたデータを読み込む命令です。また、DATAとは、READで読み込むデータを用意する命令です。このように、READとDATAは、いつも対にしていっしょに使用します。なお、DATAはプログラム中どこに配置してもかまいません。

X～Fまでのデータ（マリオの表示位置の座標や各種マリオの番号）を読み込みます。

```
READ X,Y,A,B,C,D,E,F
  ⁝
DATA 120,140,1,3,0,2,4,5
```

READで読み込まれるデータ



READのあとには、変数（数値変数名、または文字変数名）を書き、DATAのあとには、それに対応する定数データ（数値変数値、または文字変数値）を書きます。だから、READの変数とDATAの定数データは、1対1で対応し、かつ双方同じ型でなければなりません。

※68ページをご覧ください。

●PAUSE（ポーズ）について…

PAUSEとは、プログラムの実行を一時休止する命令です。

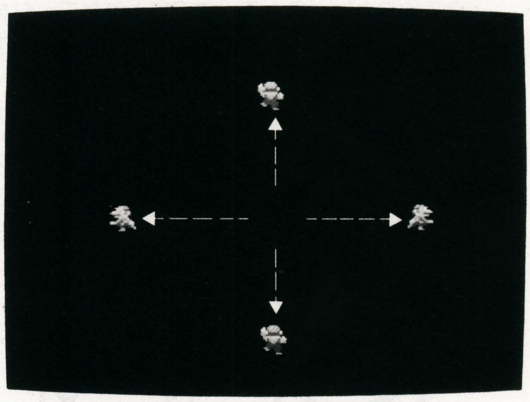
PAUSE　n
└プログラムの実行を休止させる時間（0～32767で指定します）

n（プログラムの実行を休止させる時間）を省略した場合、何かキーが押されるまでこのPAUSEの所でプログラムの実行が中断する機能があります。何かキーが押されると、PAUSEの次のプログラムから実行が再開されます。

※78ページをご覧ください。

マリオを上下左右へ自由に動かす場合には、28ページのプログラム5を使い、次のプログラムのうち行番号60〜130, 3000〜3020を入力してください。

```
5 CLS
10 SPRITE ON
15 CGSET 1,0
20 DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+CHR$(0)
                           +CHR$(3)+CHR$(2)
21 DEF SPRITE 1,(0,1,0,0,0)=CHR$(0)+CHR$(1)
                           +CHR$(2)+CHR$(3)
22 DEF SPRITE 2,(0,1,0,1,0)=CHR$(5)+CHR$(4)
                           +CHR$(7)+CHR$(6)
23 DEF SPRITE 3,(0,1,0,0,0)=CHR$(4)+CHR$(5)
                           +CHR$(6)+CHR$(7)
24 DEF SPRITE 4,(0,1,0,0,0)=CHR$(20)+CHR$(21)
                           +CHR$(22)+CHR$(23)
25 DEF SPRITE 5,(0,1,0,1,0)=CHR$(21)+CHR$(20)
                           +CHR$(23)+CHR$(22)
26 X=100
30 READ X,Y,A,B,C,D,E,F
40 DATA 120,140,1,3,0,2,4,5
50 SPRITE A,X,Y
60 S=STICK(0):IF STRIG(0)<>0 THEN END
65 IF S=0 THEN 60
70 IF S>2 THEN 100
80 X=X+2:IF S=2 THEN X=X-4
90 GOTO 101
100 Y=Y+2:IF S=8 THEN Y=Y-4
101 IF X>255 THEN X=X-252
102 IF X<3 THEN X=X+252
103 IF Y>240 THEN Y=Y-237
104 IF Y<3 THEN Y=Y+237
110 IF S=1 THEN 1000
120 IF S=2 THEN 2000
130 IF S>2 THEN 3000
1000 PAUSE 5:SPRITE C,X,Y:SWAP C,D
1010 SPRITE A:SPRITE B:SPRITE C:SPRITE E:SPRITE F
1020 GOTO 60
2000 PAUSE 5:SPRITE A,X,Y:SWAP A,B
2010 SPRITE A:SPRITE C:SPRITE D:SPRITE E:SPRITE F
2020 GOTO 60
3000 PAUSE 5:SPRITE E,X,Y:SWAP E,F
3010 SPRITE A:SPRITE B:SPRITE C:SPRITE D:SPRITE E
3020 GOTO 60
```

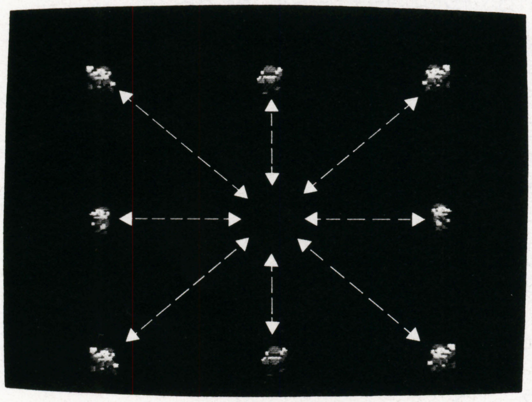


このプログラムは、28ページのプログラム5を基本に使い、上下方向のマリオの動き、画面からマリオがはみ出さないようにする処理、プログラムを終了させる処理などを追加したものです。
上下方向のマリオの動きや表示は、行番号100、130、3000〜3020で行なっています。
マリオが画面からはみ出ないように、行番号101〜104で行なっています。
また、行番号60〜65では、コントローラのトリガーボタンを押してプログラムを終了させる処理をSTRIG命令で行なっています。

■MOVE(ムーブ) コマンドについて

MOVEコマンドは、ファミリーベーシックの特長となるコマンドです。さあ、簡単なMOVEコマンドを使って、マリオやレディなどのアニメキャラクタをダイナミックに動かしてみましょう。

★マリオを同時に、8方向に動かそう



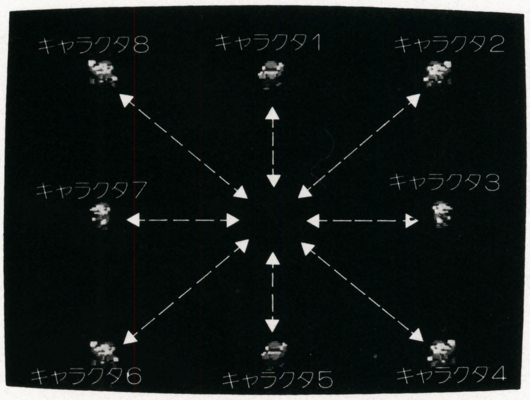
NEW RETURN

を入力し、前のプログラムをメモリから消しておきます。

```
5 CLS RETURN
10 SPRITE ON RETURN
20 CGSET 1,0 RETURN
30 FOR N=0 TO 7 RETURN
40 DEF MOVE(N)=SPRITE(0,N+1,3,
   255,0,0) RETURN
50 NEXT RETURN
60 MOVE 0,1,2,3,4,5,6,7 RETURN
70 GOTO 60 RETURN
```

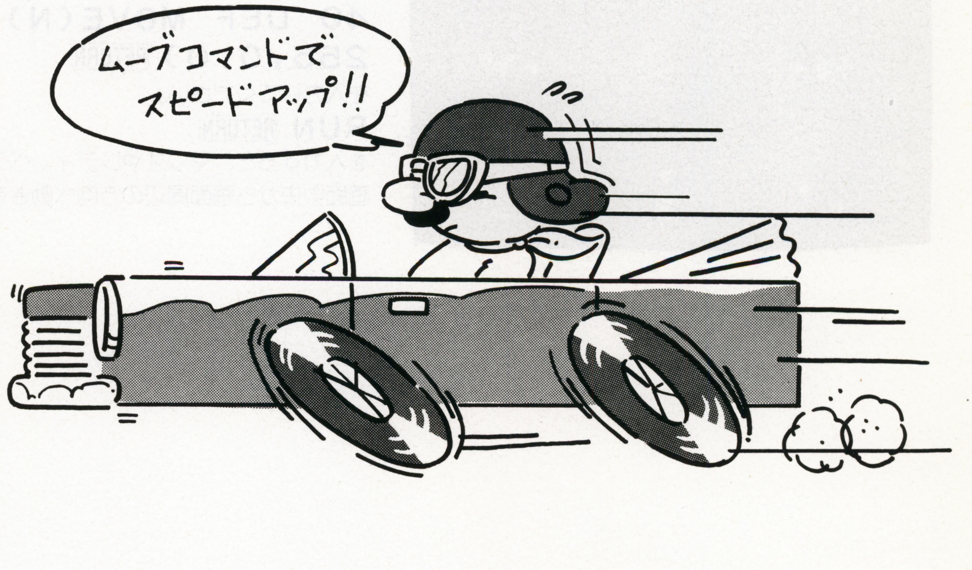
プログラム6

を入力してください。



RUN RETURN

を入力すると、8人のマリオがそれぞれ8つの方向に動きます。



●DEF MOVE（デフ ムーブ）について…

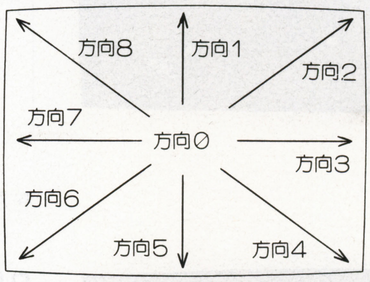
DEF MOVEとは、アニメキャラクタの種類と動きを定義する命令です。

DEF MOVE(n)=SPRITE(A,B,C,D,E,F)

- n：アニメキャラクタの動作番号(0～7)
- A：アニメキャラクタの種類(0～15)　全部で16種類のアニメキャラクタを指定できる。

0：マリオ	8：スターキラー
1：レディ	9：スターシップ
2：ファイターフライ	10：爆発
3：アキレス	11：ニタニタ
4：ペンペン	12：レーザー
5：ファイアーボール	13：シェルクリーパー
6：車	14：サイドステッパー
7：スピナー	15：ニットピッカー

- B：アニメキャラクタの移動方向指定(0～8)（0は静止）
- C：アニメキャラクタを動かす速さ(1～255)
  - 1：最高速
  - 255：最低速
- D：アニメキャラクタを移動させる全移動量(1～255)（0を指定すると表示されません）
- E：表示の優先度(0～1)
  - 0：バックグラウンドより手前にアニメキャラクタを表示する。
  - 1：バックグラウンドより後にアニメキャラクタを表示する。
- F：アニメキャラクタの配色番号(0～3)
  ※113ページのカラーチャートをご覧ください。



※74ページをご覧ください。

●MOVE（ムーブ）について…

MOVEとは、あらかじめDEF MOVE命令で定義されたアニメキャラクタの動きを開始させる命令です。

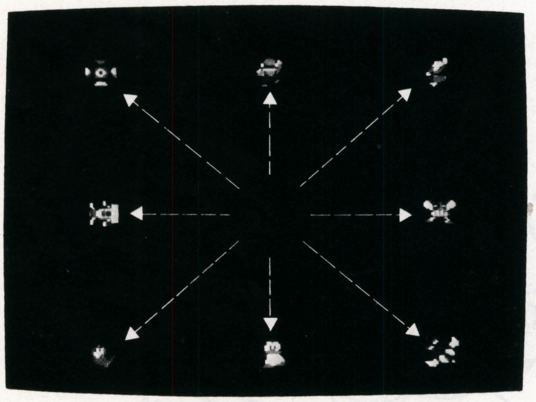
MOVE 0,1,2,3,4,5,6,7

MOVE命令は、同時に8個までのアニメキャラクタの動きを開始させることができます。アニメキャラクタは、MOVE命令が実行されると、DEF MOVE命令のD(全移動量)の値で指定された移動量が終わるまで動きます。
動作番号0～7までのアニメキャラクタのうち、希望の番号のアニメキャラクタのみを動かすことができます。

MOVE 1……………動作番号1番のアニメキャラクタのみを動かす。
MOVE 0, 3, 6…動作番号0番，3番，6番のアニメキャラクタを同時に動かす。

※75ページをご覧ください。

★同時に8種類のアニメキャラクタを、8方向に動かそう



マリオだけのように、1種類のキャラクタだけでなく、8種類のキャラクタを同時に動かすことができます。

31ページのプログラム6を使って

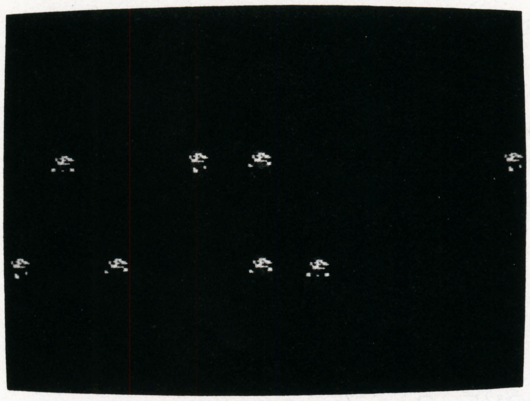
40 DEF MOVE(N)=SPRITE(N,N+1,3,255,0,0) RETURN

を入力してください。

RUN RETURN

を入力すると、マリオやレディ、ペンペンなど、8種類のアニメキャラクタが、画面中央から画面周辺の方向へ動きます。

★8人のマリオを横に行進させよう

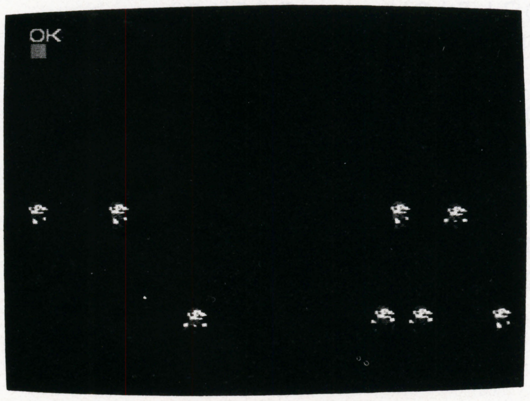


31ページのプログラム6を使って、

```
40 DEF MOVE(N)=SPRITE(0,3,N+1,255,
    0,0) RETURN
55 FOR N=0 TO 3:POSITION 2*N+1,120,
    180:NEXT RETURN
70 END RETURN
RUN RETURN
```

を入力してください。
8人のマリオが左から右へ歩きます。

★マリオよ、とまれ!



同時に動いているキャラクタを止めたり、ある特定のキャラクタだけ止めたりすることができます。
上のプログラムに、

```
70 PAUSE RETURN
80 CUT 0,1,2,3,4,5,6,7 RETURN
90 END RETURN
RUN RETURN
```

を入力してください。

8人のマリオが歩き出したら、SPACEキーを一度押してください。SPACEキーが押されると同時に、行番号80の命令が実行され、歩いていた8人のマリオが同時にピタリと止まります。
ここで、

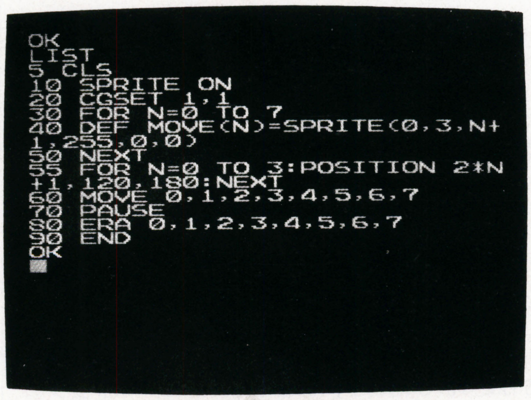
```
MOVE 0,2 RETURN
```

を入力すると、止まっていた動作番号0と2のマリオが歩き出します。
さらに、

```
80 CUT 0,3 RETURN
RUN RETURN
```

を入力してください。
8人のマリオが歩き出したら、SPACEキーを一度押してください。動作番号0と3のマリオだけが止まり、他のマリオはなにもなかったように、左から右へ歩き出します。
CUT命令は、ダイレクトモード、プログラムモードの両方で使用できます。

★あっ!マリオが消えたゾ



よりゲームを面白くするには、アニメキャラクタを画面から消すことも重要なテクニックです。同時に8種類のキャラクタを消したり、特定のキャラクタだけを消すことができます。

```
80 ERA 0,1,2,3,4,5,6,7 RETURN
RUN RETURN
```

を入力してください。
8人のマリオが表示され、歩き出したら、SPACEキーを一度押してください。
行番号80の命令が実行され、歩いていた8人のマリオが消えてしまいます。
ここで、

```
MOVE 0,1 RETURN
```

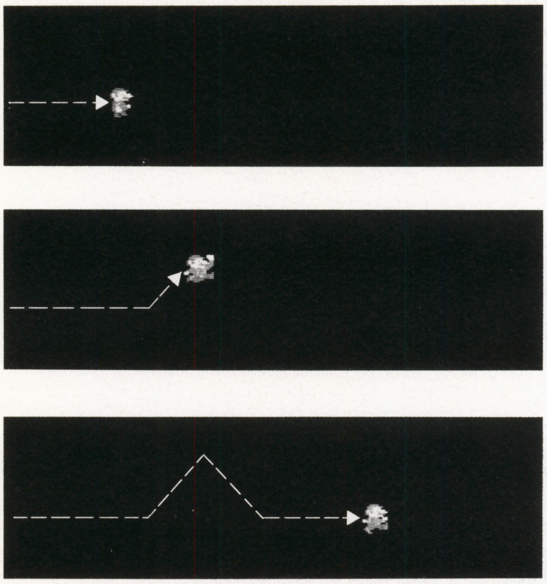
を入力すると、動作番号0と1のマリオが登場し、歩き出します。
さらに、

```
80 ERA 3,4 RETURN
RUN RETURN
```

を入力してください。
8人のマリオが歩き出したら、SPACEキーを一度押してください。動作番号3と4のマリオだけが消え、他のマリオはなにもなかったように、左から右へ歩きます。
ERA命令は、ダイレクトモード、プログラムモードの両方で使用できます。

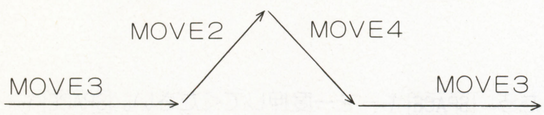
> CUT(カット)やERA(イレイズ)命令の後で、MOVE命令を実行すると、アニメキャラクタはCUT、またはERAされた位置からDEF MOVE命令で定義されたDの値(全移動量)の残った値を実行し、停止します。

★マリオをジャンプさせよう



マリオがジャンプしているように見せるには、次のプログラムを入力してください。

```
5 CLS
10 SPRITE ON
20 CGSET 1,0
30 FOR N=2 TO 4
40 DEF MOVE(N)=SPRITE(0,N,1,20,1,0)
50 NEXT
60 MOVE 3
70 IF MOVE(3)=-1 THEN 70
80 ERA 3:POSITION 2,XPOS(3),
   YPOS(3):MOVE 2
90 IF MOVE(2)=-1 THEN 90
100 ERA 2:POSITION 4,XPOS(2),
    YPOS(2):MOVE 4
110 IF MOVE(4)=-1THEN 110
120 ERA 4:POSITION 3,XPOS(4),
    YPOS(4):GOTO 60
```



もし、マリオの動きに合わせて、効果音を入れる場合は、

```
65 PLAY "T104C1B1DEG1CDE1"
85 PLAY "O3CDE1G1A"
```

を追加入力してください。

RUN RETURN

を入力すると、左から歩いてきたマリオは画面中央でジャンプし、着地すると、再び歩きだします。

●MOVE(n)について…（ムーブ）

MOVE(n)とは、MOVE文で動かしたアニメキャラクタの動きが実行中か停止中かを求める命令です。

MOVE(n)……… {・指定のアニメキャラクタが実行中ならば、−1が求められます。 ・指定のアニメキャラクタが停止中ならば、0が求められます。}

n：アニメキャラクタの動作番号(0〜7)

```
70 IF MOVE(3)=-1 THEN 70
80 ERA 3:POSITION 2,XPOS(3),YPOS(3):MOVE 2
```

このプログラムでは、動作番号3番のアニメキャラクタの動きが実行中であれば、行番号70の命令を実行し、そうでなければ、行番号80の命令を実行します。

※77ページをご覧ください。

●POSITIONについて…（ポジション）

POSITIONとは、MOVE文でアニメキャラクタを動かす前に、動作を開始する初期座標を与える命令です。

POSITION n,X,Y

- Y：垂直方向の座標(0〜255)
- X：水平方向の座標(0〜255)
- n：アニメキャラクタの動作番号(0〜7)

※76ページをご覧ください。

●XPOS、YPOSについて…（エックスポス、ワイポス）

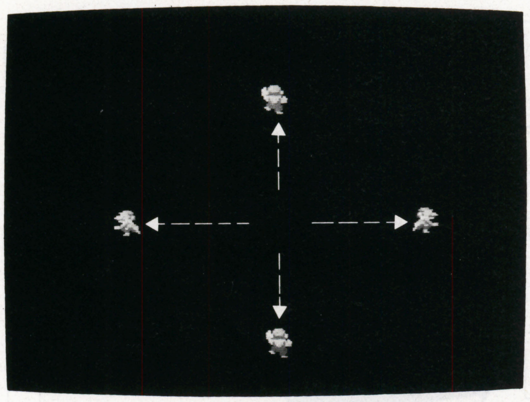
XPOSとは、アニメキャラクタの動作番号の水平方向の座標値を求める命令です。また、YPOSとは、アニメキャラクタの垂直方向の座標値を求める命令です。

XPOS(n),YPOS(n)

n：アニメキャラクタの動作番号(0〜7)

※76ページをご覧ください。

★マリオをコントローラで動かそう



さて、Ⅰコントローラを使用して、マリオを上下左右に動かしてみましょう。
次のプログラムを入力してください。

```
5 CLS
10 SPRITE ON
20 CGSET 1,0
30 FOR N=0 TO 7
40 DEF MOVE(N)=SPRITE(0,N,1,3,0,0)
50 NEXT
60 S=STICK(0)
70 IF S=0 THEN N=0:GOTO 120
80 IF S=1 THEN N=3:GOTO 120
90 IF S=2 THEN N=7:GOTO 120
100 IF S=4 THEN N=5:GOTO 120
110 IF S=8 THEN N=1
120 IF MOVE(M)=-1 THEN 120
130 IF M=N THEN 160
140 ERA M:POSITION N,XPOS(M),
  YPOS(M):M=N
150 MOVE N:GOTO 60
160 MOVE M:GOTO 60
```

RUN RETURN

を入力すると、Ⅰコントローラを使って、マリオを自由に動かすことができます。
30ページのプログラムと比較してわかるように、同じ動作をさせるにも、MOVEコマンドを使用すると、プログラムは短くてすみますね。

●MOVE（ムーブ）コマンドの特長

①MOVEコマンドのプログラムを実行中に、STOPキーを押して、プログラムリストを表示したり、プログラムを修正することができます。

②MOVEコマンドを実行中は、演奏のテンポが遅く（½）なります。

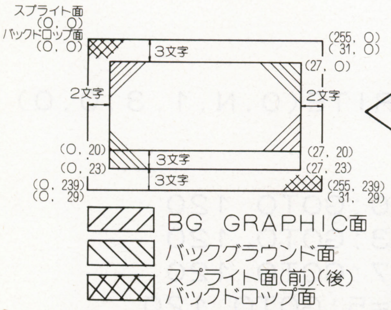
③MOVEコマンドで動きを指示できるキャラクタは、同時に8種類です。
ただし、アニメキャラクタは、横方向には同時に4個までしか表示しません。

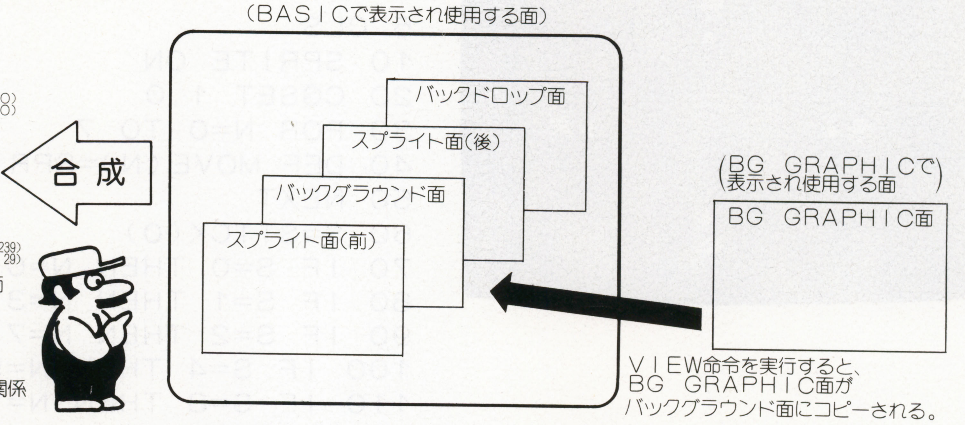
④MOVEコマンドで扱えるキャラクタは16種類です。

⑤DEF SPRITE、SPRITE命令と合わせて使えば、16種類のアニメキャラクタを動かすことができます。

■表示画面のしくみ

下図はファミリーベーシックの画面を前から見たものです。



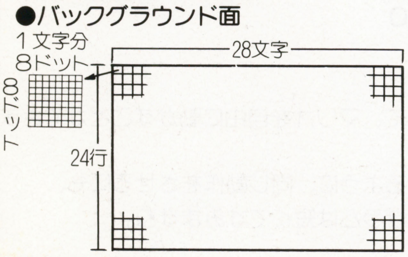


上の図は、それぞれの画面の相対的な位置関係を示しています。
上の数字は、それぞれの面で指定できる座標値を示しています。

BASICの表示画面は、スプライト面(前)、バックグラウンド面、スプライト面(後)、バックドロップ面の4面で構成されています。(バックグラウンド面の前方や後方にスプライト面を設定することができます) この他にBG GRAPHICを描くBG GRAPHIC面があります。

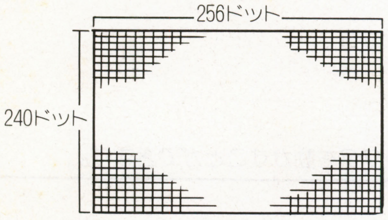
各表示画面の説明

●バックグラウンド面



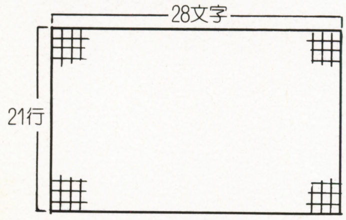
- **GAME BASICモードで、常に表示されている面で、BASICプログラムを作成する面です。**(大きさはタテ24行、ヨコ28文字です)
- キーボードから入力した英数字やカナ、記号が表示されます。
- GAME BASICモードでは、この面を中心にして、BASIC命令でスプライト面をバックグラウンド面の前や後に表示することができます。
- BG GRAPHIC面の内容をVIEW命令でこの面にコピー(複製)することができます。

●スプライト面(前)、(後)



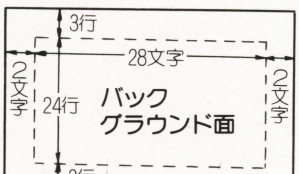
- **マリオなど動きのあるアニメキャラクタ(画面上ではスプライトと呼びます)を表示する面です。**
- この面上にスプライトを表示する位置は、タテ240ドット、ヨコ256ドットの単位で指定できます。(テレビ画面上への表示範囲は、テレビ受像機により異なります。)
- BASICの命令でこの面を呼び出したり、消したりできます。この場合、この面上にあるすべてのスプライトが見えたり、消えたりします。
- バックグラウンド面の前と後にありますから、どちらの面にスプライトを表示させるかによって、スプライトとバックグラウンド面の英数字やカナ、記号、グラフィックキャラクタとの間で優先順位(バックグラウンド面にアニメキャラクタを隠したり、現わしたりすることができる)がつけられ、遠近表現ができます。

●BG GRAPHIC面



- **BG GRAPHICを描く面です。**(大きさはタテ21行、ヨコ28文字です)
- GAME BASICモードで、VIEW命令を使って、バックグラウンド面に、この面に描かれた画面の内容を呼び出してコピー(複製)することができます。
- GAME BASICモードで、BG GRAPHICと同じ絵をバックグラウンド面に表示して、スプライト面上でアニメキャラクタを動かせば、ゲーム表現に工夫ができます。
- バックグラウンド面にコピー(複製)したBG GRAPHIC面の画面はBASICの命令で自由に修正できますが、BG GRAPHIC面には元の絵が保存されています。

●バックドロップ面



- **常に表示されているバックグラウンド面の背景色を表示する面です。**
- バックグラウンド面より大きく、タテ30行、ヨコ32文字の大きさの面です。
- 着色は、この画面全体に、1色で表現できます。初期設定は黒(透明)になっています。
- この面は空や海などの表現に便利です。

■コンピュータ音楽や、ゲームの効果音

ファミリーベーシックには、3つの音を同時に出して自動演奏したり、アニメキャラクタの動きに合わせて効果音を出すなど楽しい音楽機能があります。
この機能をフルに活用して、演奏してください。

★音楽演奏にチャレンジしよう

1 コンピュータで「メリーさんの羊」を演奏してみましょう。

NEW RETURN
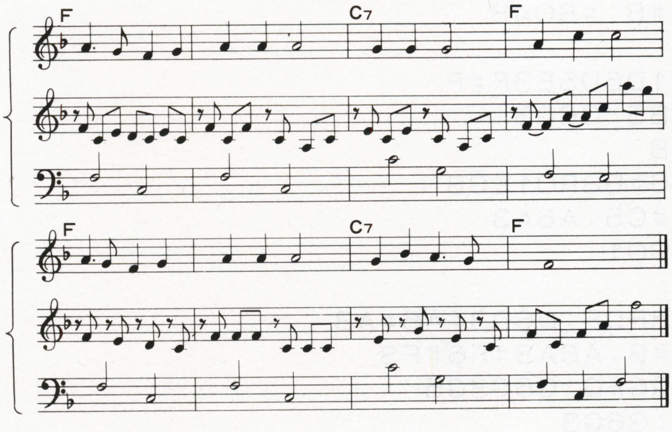
を入力し、前のプログラムをメモリから消します。
次のプログラムを入力してください。

```
5 CLS
6 LOCATE 8,12:PRINT "メリーサン ノ ヒツジ"
10 PLAY"M1Y2V7T3:M1Y1V5T3:M1T3"
20 PLAY"O2A6G3F5G:O2R3FCEDCEC:O1F7C"
30 PLAY"A5AA7:RFCFCFRCO1AO2C:FC"
40 PLAY"G5GG7:RECERCO1GO2C:O2CO1G"
50 PLAY"A5O3CC7:RF5AO3C3AG:FC"
60 PLAY"O2A6G3F5G:O2RFRERDRC:FC"
70 PLAY"AAA7:RFFFRCCC:FC"
80 PLAY"G5#AA6G3:RERGRERC:O2CO1G"
90 PLAY"F9:FCFAO3F7:F5CF7"
```

RUN RETURN

を入力すると、コンピュータは「メリーさんの羊」の曲を自動演奏しますね。

「メリーさんの羊」





2 松田聖子ちゃんの「ROCKN ROUGE」を演奏してみましょう。

NEW RETURN

を入力し、前のプログラムをメモリから消します。
または、SAVE(保存)しておいてください。
次のプログラムを入力してください。

```
10 PLAY "M1V1ØY2T3:M1V7YØT3:M1T3
2Ø PLAY "O4G6C7:O3C3C5C3C5C3C:O2C7C
3Ø PLAY "R7G1A3GFE1:RCC5CC:O1#A#A
4Ø PLAY "F4ED3C7:C3C5C3C5C3C:AA
5Ø PLAY "F4#DD3C5D:O2#G#B#B#G#BG#BF:#GG
6Ø PLAY "O4G6C7:O2#B3#BG#B#BG#B#B:O2CC
7Ø PLAY "R7G1A3GFE1:F#B#BF#BF#BF:O1#A#A
8Ø PLAY "F4ED3C7:#B#BA#B#BA#B#B:AA
9Ø PLAY "F4#DD3:R3F1G#G#AO3CD:#G6#G3
95 PLAY "#D5F3O3C:C5D3G:#G5#A
97 N=Ø
1ØØ PLAY "RDECG5C3C:O3:O1#B6#B3#B5G
11Ø PLAY "RDEC:R7:#A6#A3
115 PLAY "G6C3:F1ED#CC#CD#C:#A5F
```

次のページにつづきます。

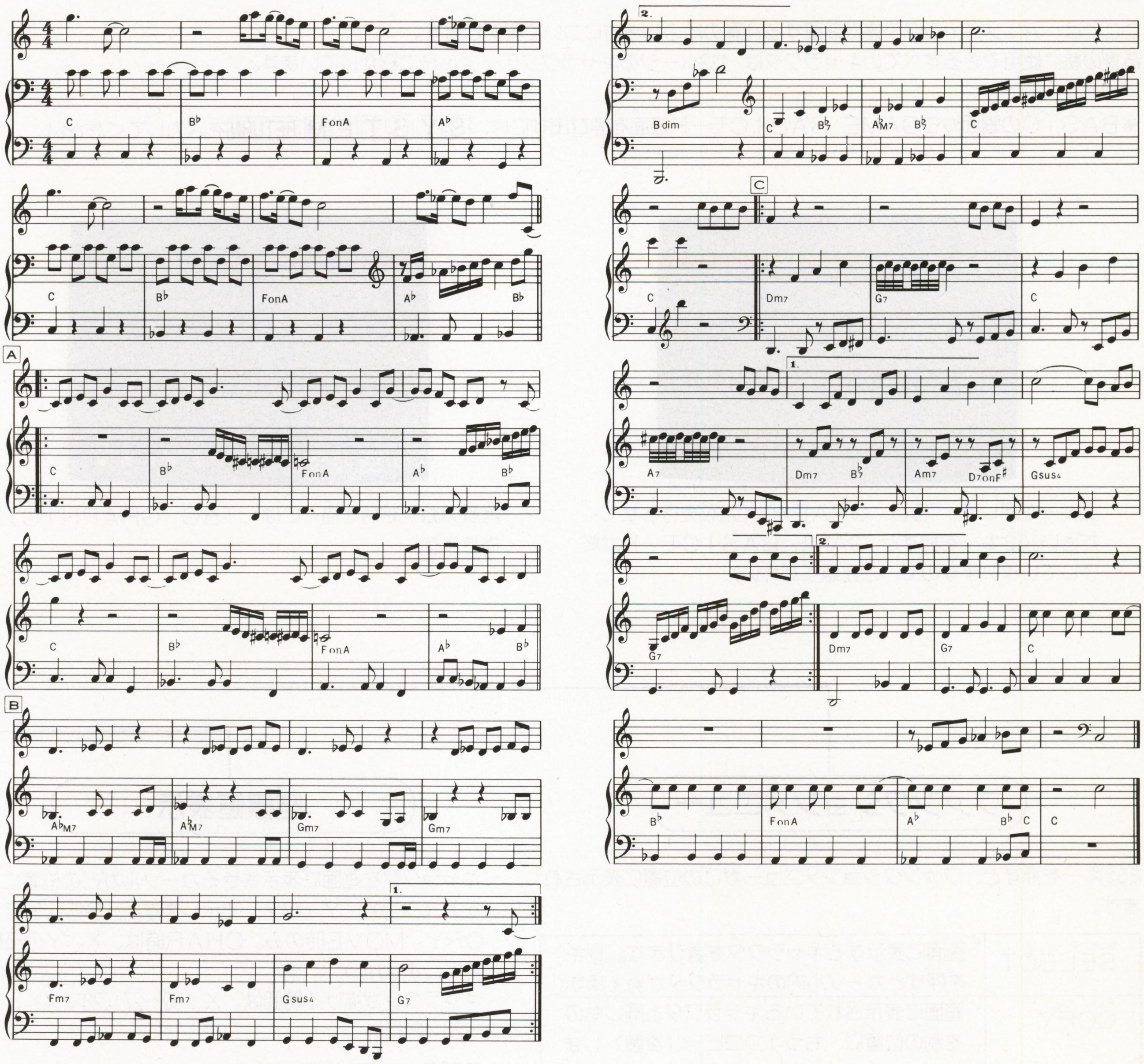
```
120 PLAY "RDECG5C3G:C7:A6A3A5F
130 PLAY "RGFC:R7:#G6#G3
135 PLAY "CDRC:F1GA#AO4CDEF:#G5#A3#B
140 PLAY "RDECG5C3C:G7O3:O1#B6#B3#B5G
150 PLAY "RDEC:R7:#A6#A3
155 PLAY "G6C3:F1ED#CC#CD#C:#A5F
160 PLAY "RDECG5C3G:C7O2:A6A3A5F
170 PLAY "RGFCC5D:R7#D5F:#B3#B#A#G#G5#A
180 PLAY "D6#D3:O2#A6#B3:#G5#G
185 PLAY "#D5:O3C5C3D:#G#G3#G1#G
190 PLAY "R5D3#D:#D5:#G5#G
195 PLAY "D#DF#D:R5C3C:#G5#G3#G
200 PLAY "D6#D3:O2#A6#B3:G5G
205 PLAY "#D5:#B5G3A:GG3G1G
210 PLAY "R5D3#DD#DF#D:#A8O2#A3#A:O1G5GGG
220 PLAY "F6G3G7:O3D6#D3#D7:F5F3G#G5#G3G
230 PLAY "F5G#DF:D5#DCD:F5F3G#G5F
240 PLAY "G8:O3BO4CDCO3:GGG3EDO0B
245 IF N=1 THEN 260
250 PLAY "R8R3C:B7G1ABO4CDEFG:O1G5GG3AB#B
255 N=1:GOTO 100
260 PLAY "#G5GFD:O1R3DF#GB7:O0B5
270 PLAY "F6#D3#D7:O2G5O3CD#D:O1#B#B#A#A
280 PLAY "F5G#G#A:D#DFG:#G#G#A#A
290 PLAY "#B8:O3C1DEFG#GABO4CDEF:#B#B#B
295 PLAY "R5:G#GAB:#B
300 PLAY "R7#B3B#BB:O4#B5#B:#BO4B
305 M=0
310 PLAY "O3F5:R5O3FA#B:O1D6D5E3F#F
320 PLAY "R:B0#BB#BB#BB#BB5:G6G3
325 PLAY "#B3B#BB:R7:R3GAB
330 PLAY "O3E5:R5GBO4D:#B6#B3RO1EGB
340 PLAY "R:#C0D#CD#CD#CD#C5:A6A3
345 PLAY "A3GAG:O3R:R3GE#CO1
347 IF M=1 THEN 390
350 PLAY "C5C3FD5D3G:R3FARRDF:D6D3#A6#A3
360 PLAY "E5AB#B:RCERRO2A#B:A6A3#F6#F3
370 PLAY "#B7R3BAB:O3RCDFGGFG:G6G3G5
380 PLAY "R:O2G1O3CDFDFGB:G6G3
385 PLAY "#B3B#BB:GBO4DFDFGB:G
387 M=1:GOTO 310
390 PLAY "F5F3GF5F3G:D5D3ED5D3E:D7O1#A5A
400 PLAY "F5A#BB:D5FGF:G6G3G6G3
410 PLAY "#B8:O4C3C5C3C5C3C:#B5#B#B#B
420 PLAY ":RCC5CC:#A#A#A#A
430 PLAY ":C3C5C3C5C3C:AAAA
440 PLAY "R3#DFG#G5:RCC5C:#G#G#G
445 PLAY "#A3#B:C3C:#A3#B
450 PLAY "O1R7C:R7C
```

RUN RETURN

を入力すると、コンピュータは「ROCKN ROUGE」を演奏します。

Rock'n Rouge（ロックル・ルージュ）

呉田軽穂(作)



★マリオの動きに合わせて音を出そう。

ゲームをおもしろくするには音楽をつけるのが効果的です。
次のプログラムを入力してください。

```
5 CLS:SPRITE ON:CGSET 1,0
10 FOR N=0 TO 2:DEF MOVE(N)=SPRITE(0,N+2,1,20,1,0):NEXT
20 FOR K=3 TO 7:DEF MOVE(K)=SPRITE(8+K,7,1,255,1,0):NEXT
30 POSITION 7,0,30:MOVE 1
40 PLAY"T104C1D1G1B1CGAB"
50 IF MOVE(1)=-1 THEN 50
60 ERA 1:POSITION 0,XPOS(1),YPOS(1):MOVE 0,6
70 PLAY"T105CDEABGD"
80 IF MOVE(0)=-1 THEN 80
90 ERA 0:POSITION 2,XPOS(0),YPOS(0):MOVE 2,5
100 IF MOVE(2)=-1 THEN 100
110 ERA 2:POSITION 1,XPOS(2),YPOS(2):MOVE 7,3:GOTO 30
```

RUN RETURN

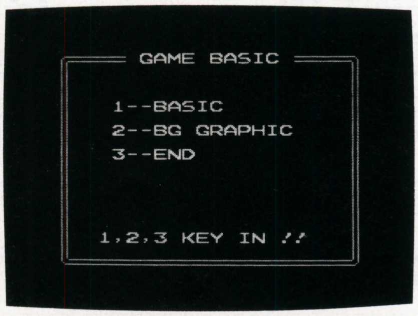
を入力すると、画面のマリオの動作に合った音が出るので、いっそう楽しくなります。

※PLAY命令の説明は、80ページをご覧ください。また、LOCATE命令の説明は70ページをご覧ください。

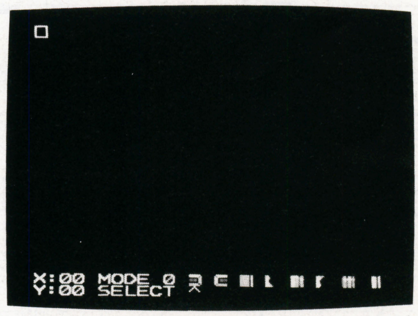
BG GRAPHIC

ここでは、アニメキャラクタ達が活躍する背景の絵の描き方について説明します。
背景の絵に使用できるすべてのキャラクタは 113ページのキャラクタテーブルBで紹介しています。

※BASICの画面からGAME BASICモード画面を呼び出すには、SYSTEM RETURNを入力してください。



2キーを押して、「BG GRAPHIC」を実行させてください。(3キーを押すと、GAME BASICモードが終了してしまいますので、ご注意ください)



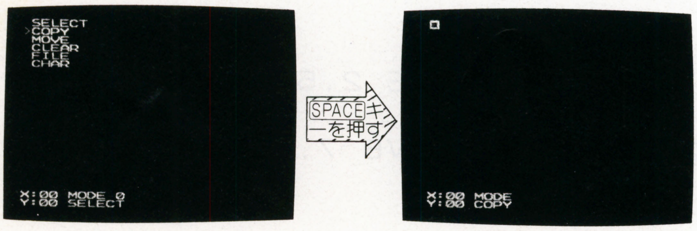
背景の絵や絵柄を描くには、「BG GRAPHIC」で描きます。

ファンクションメニュー

ESCキーを押すと、ファンクションメニューがこの位置に表示されます。

SELECT	画面に表示するキャラクタを選びます。Dキーを押すとカーソル内のキャラクタが消えます。
COPY	画面に表示されているキャラクタと同じものを別の位置に、もう1つコピー（複製）します。
MOVE	画面に表示されているキャラクタを別の位置に移動します。
CLEAR	画面に表示されているすべてのキャラクタを消去します。
FILE	画面に表示されている絵をカセットテープにSAVE（保存）したり、カセットテープから絵のデータをLOAD（呼び出し）します。※46ページをご覧ください。
CHAR	英数字、記号、カナを表示します。

ファンクションを選ぶには、▼キーを押します。
SPACEキーを押すと、選んだファンクションのメニューが実行されます。



座標値表示

キャラクタを画面に表示させるカーソルがどの位置にあるかをX（ヨコ）、Y（タテ）で示します。（SELECT、COPY、MOVE時のみ。CHAR時は、X、Yの座標値は変化しません。）
キャラクタが描ける範囲は、X：00～27（28マス）、Y：00～20（21行）

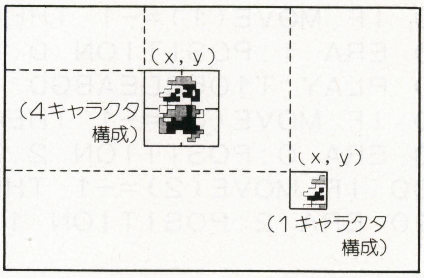
スプライト面の座標とBG GRAPHIC面の座標の関係

x…スプライトのx座標
y…スプライトのy座標
X…BG GRAPHICのX座標
Y…BG GRAPHICのY座標

スプライト面とBG GRAPHIC面の座標を上のように定めると、背景の絵にアニメキャラクタを合成させるときの座標は、次のような関係の式になります。

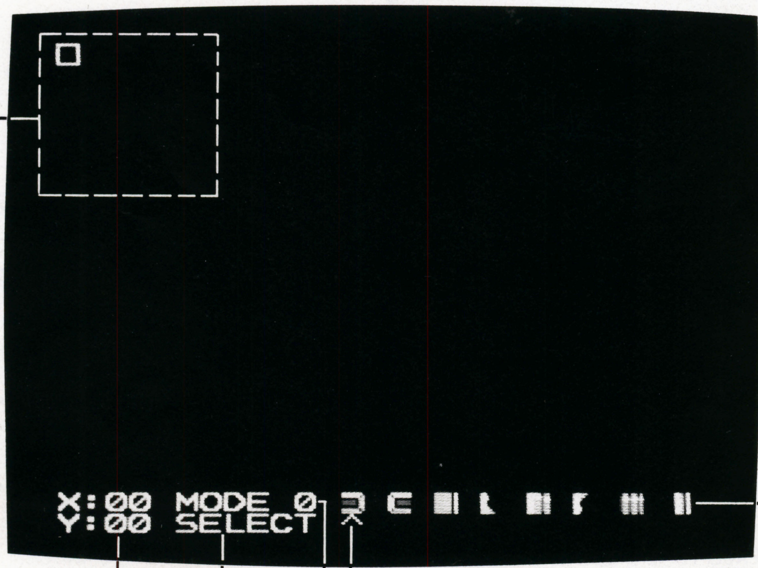
x＝（X×8）+16　y＝（Y×8）+24

従って、背景の絵柄にアニメキャラクタを重ねて表示したい場合は、あらかじめ、上の式を考慮してプログラムを作ってください。



※36ページの「表示画面のしくみ」もご覧ください。

▲▼◀▶キーで、キャラクタを表示する座標値にカーソル（□）を動かします。



キャラクタグループ

1グループは、8個のキャラクタで構成されています。
CLR HOMEキーを押すと、グループが順方向に変わります。
13グループあります。
SHIFTキーを押しながらCLR HOMEキーを押すと、グループは逆方向に変わります。
キャラクタは、8×13=104種類用意されています。
（113ページのキャラクタテーブルBをご覧ください）

実行モード表示

ファンクションメニューで選択したモードが表示されます。

配色番号

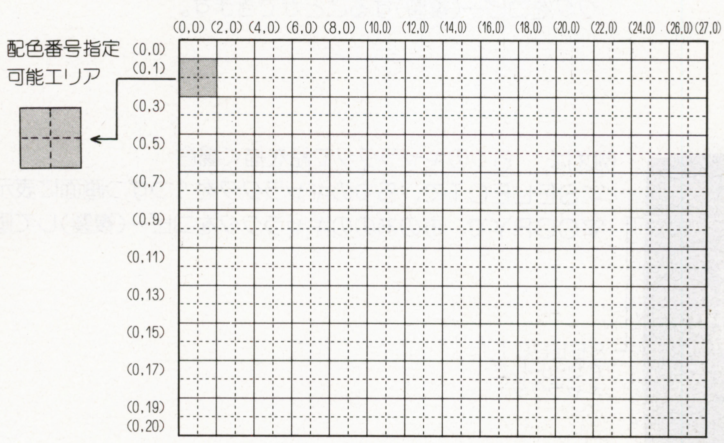
RETURNキーを押すごとに、キャラクタの色が変わります。
配色は4種類（0～3）の中から選べます。

配色番号は、□のエリア内ごとに使用できます。従って、あらかじめこのエリア内に描かれてある絵を他の配色番号で描き直すと、このエリア内の色がすべて変わります。

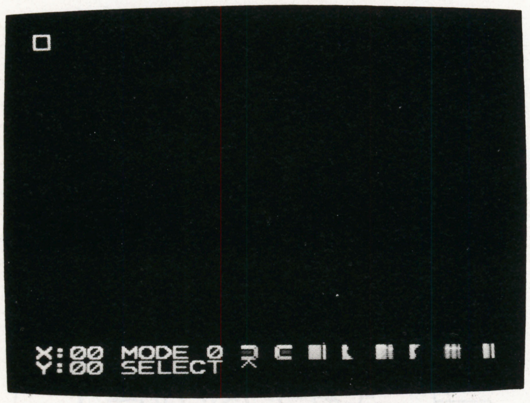
※113ページのカラーチャートをご覧ください。

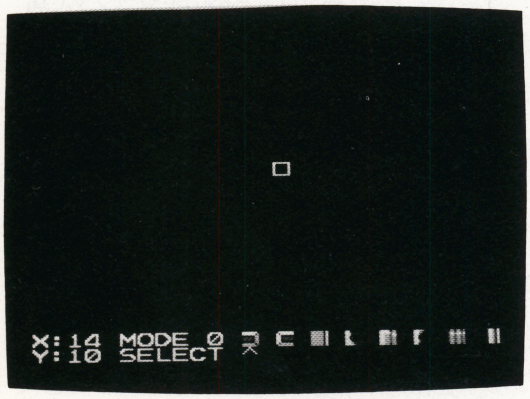
キャラクタマーカー

キャラクタグループの中のどれか1つのキャラクタを選ぶときのマーカーです。
DELキーを押すと、キャラクタマーカーは右へ移動します。
INSキーを押すと、キャラクタマーカーは左へ移動します。



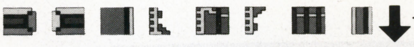
★SELECTで背景や絵を描こう





SELECTは画面に表示するキャラクタの種類や配色を選ぶ働きがあります。
まず、ESCキーを押して、ファンクションメニューを表示してください。

▽1 SPACEキーを押して、SELECTモードにします。
CLR HOMEキー、またはSHIFTキーを押しながらCLR HOMEキーを押して、画面右下に表示されるキャラクタのグループを選びます。

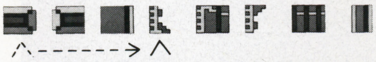


…CLR HOMEキーを押すと
順方向にキャラクタが表示されます。

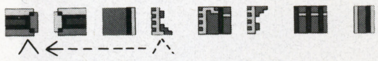


…SHIFTキーを押しながらCLR HOMEキーを押すと
逆方向にキャラクタが表示されます。

▽2 DELキー、INSキーを押して、表示させたいキャラクタを選びます。



…DELキーを押すとキャラクタマーカーが右へ移動



…INSキーを押すとキャラクタマーカーが左へ移動

▽3 RETURNキーを押して、表示させたいキャラクタの配色を変えます。配色は0〜3の4種類あります。
▽4 ▲▼◀▶キーを押して、表示させたい位置にカーソルを移動します。
（例えば、X：14、Y：10にカーソルを移動します）

▽5 SPACEキーを押すと、選んだキャラクタが画面のカーソルの位置に表示されます。連続して表示させたい場合は、SPACEキーを押し続けてください。
SPACEキーは、選んだキャラクタを画面に表示させる働きがあります。

▽6 Dキーを押すと、カーソルの位置にあるキャラクタが消えます。

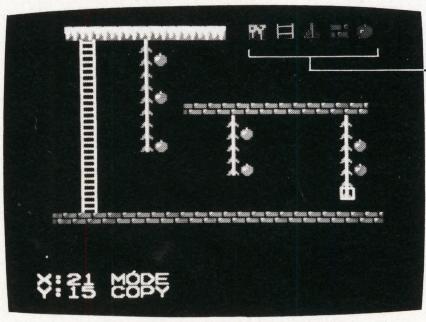
★COPYで同じパターンのキャラクタを表示しよう



COPYは表示されているキャラクタと同じものを別の位置にコピー（複製）する働きがあります。
まず、ESCキーを押してファンクションメニューを表示してください。

▽1 ▼キーを押し、COPYを選びます。
▽2 SPACEキーを押して、COPYモードにします。▲▼◀▶キーを押して、コピー（複製）したいキャラクタの位置に移動させます。（例えば、X：14、Y：10の位置にあるキャラクタ）
▽3 INSキーを押すと、そのキャラクタがどこへでも移動できる状態になります。
▽4 ▲▼◀▶キーを押して、表示させたい位置へカーソルを移動します（例えばX：12、Y：08）。元のキャラクタは表示されたままです。
▽5 位置が決まり、DELキーを押すと、カーソルの位置にキャラクタが表示されます。同じように▲▼◀▶キーとDELキーを使って、続けてキャラクタをコピー（複製）することができます。

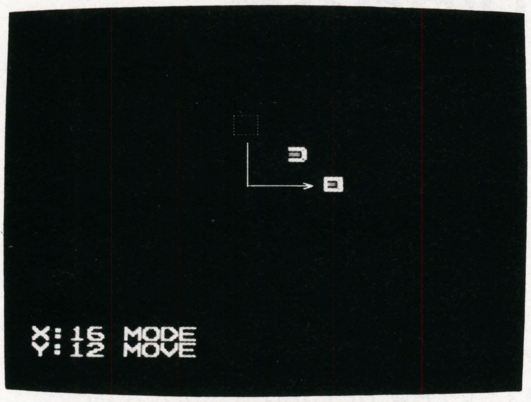
●COPYの便利な使いかた



SELECT
でセットした
キャラクタ

例えば、5つのキャラクタで絵を描く場合
▽1 SELECTで、5つのキャラクタを1つずつ画面に表示します。
▽2 COPYで、おのおののキャラクタをコピー（複製）して画面に描きます。

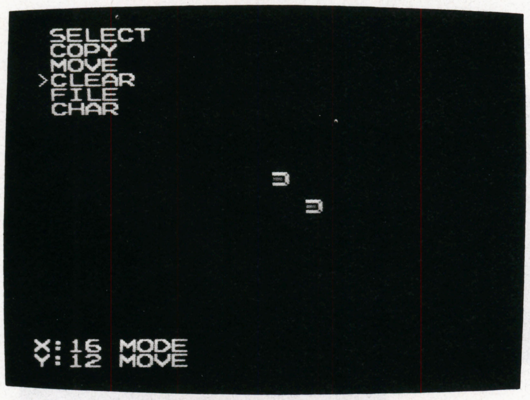
★MOVEでキャラクタの表示位置を変える



MOVEは、表示されているキャラクタを別の位置に移動させる働きがあります。誤って違う位置に表示させたキャラクタを別の位置に表示し直す場合などに便利です。
まず、ESCキーを押して、ファンクションメニューを表示してください。

▽1 ▼キーを押して、MOVEを選びます。
▽2 SPACEキーを押して、MOVEモードにします。▲▼◀▶キーを押して、移動させたいキャラクタの位置に移動します。
(例えば、X：12、Y：8)。
▽3 INSキーを1回押すと、キャラクタがとり出され、どこへでも移動できる状態になります。(INSキーを2回押すとキャラクタは消えてしまいます)
▽4 ▲▼◀▶キーを押して、改めて表示したい位置へ移動します。
(例えば、X：16、Y：12)。
▽5 位置を決め、DELキーを押すと、カーソルの位置にキャラクタが表示されます。キャラクタの移動は終りです。

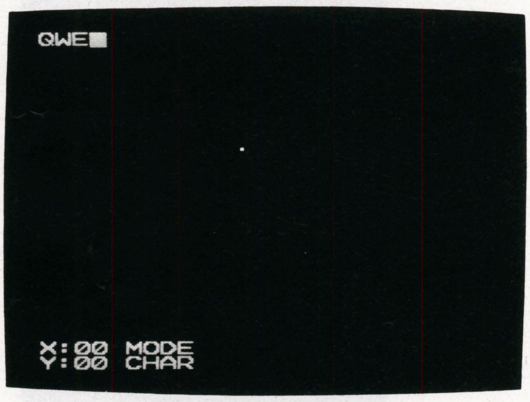
★CLEARで画面表示しているキャラクタを消す



CLEARは、画面に表示されているすべてのキャラクタを消す働きがあります。
画面に描いたキャラクタのすべてを変更したり、新たに、キャラクタを描く場合などに使用します。
まず、ESCキーを押して、ファンクションメニューを表示してください。

▽1 ▼キーを押して、CLEARを選びます。
▽2 SPACEキーを押すと、CLEARモードが実行され、画面に表示されているすべてのキャラクタが消去されます。
▽3 消した後は、自動的にSELECTになります。

★CHARで数字・英文字・記号・カナを表示する



CHARは、キーボードのもっている数字、英文字、記号、カナを表示させる働きがあります。
キャラクタグループだけでなく、数字、英文字、記号、カナが使えます。
まず、ESCキーを押して、ファンクションメニューを表示してください。

▽1 ▼キーを押して、CHARを選びます。
▽2 SPACEキーを押すと、CHARモードが実行されます。キーボードから直接カーソルの位置へ数字、英文字、記号、カナを入力します。

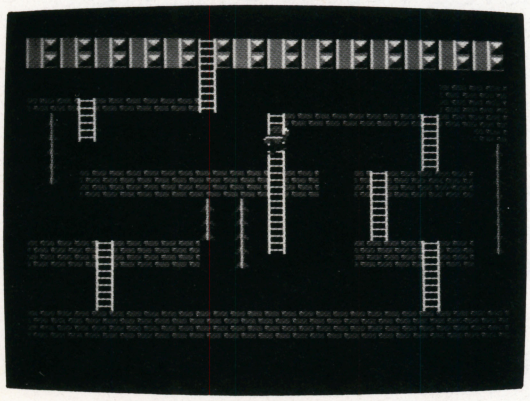
※X、Yの座標値は、カーソルを移動しても変化しません。

★FILEでカセットテープにSAVE(保存)したりLOAD(呼び出し)する ※46ページをご覧ください。

BASICとBG GRAPHICの合成

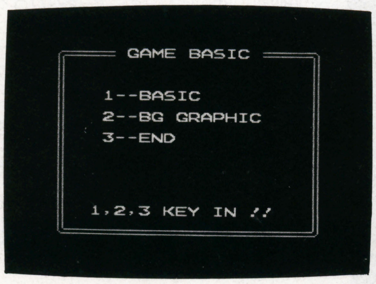
アニメキャラクタのマリオを動かすプログラムや背景の絵の作り方は、ここまでの説明で理解できたと思います。ここでは、「BASIC」で作成したプログラムと「BG　GRAPHIC」で描いた背景の絵を合成する方法について説明します。

★背景の絵とマリオを画面に表示する



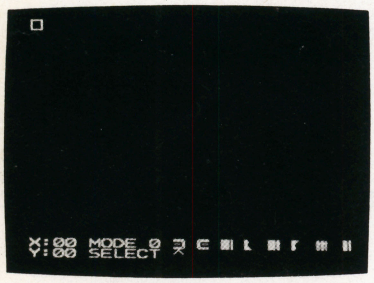
▽BG　GRAPHICで背景の絵を描いてください。
▽ESCキーを押した後、STOPキーを押し、GAME　BASICモード画面に戻します。
▽1キーを押し、BASICの画面にします。
▽下図のBASICプログラムを入力してください。
▽5　CLSのCLSに替えてVIEW RETURNを入力してください。
※他のプログラムを使われる場合、VIEW RETURNを入力する場所に注意してください。
　CLSがない………プログラムの最初の行番号より小さい行番号をつけて、VIEWを追加してください。
　CLSがある………CLSをVIEWに入れ替えます。
▽RUN RETURN
これで、アニメキャラクタと背景の絵が一緒に表示されましたね。

●マリオと背景の絵を合成するには

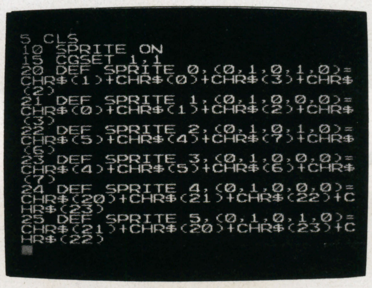


2キーを押す

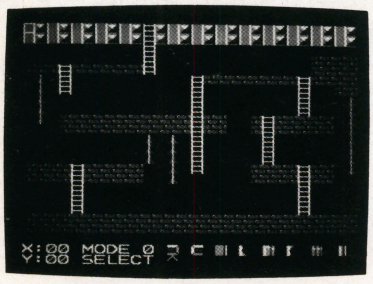
1キーを押す



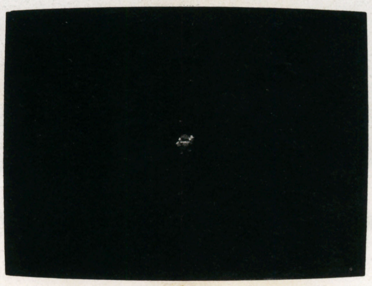
背景になる絵を描く。



マリオを表示するプログラムを作る。



背景の絵はBG　GRAPHIC面に描かれている

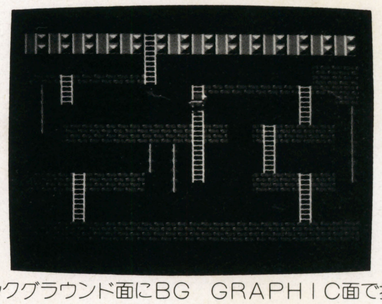


マリオはスプライト面に描かれている

VIEW命令でバックグラウンド面にBG　GRAPHIC面がコピー（複製）される

RUN RETURN

BG　GRAPHIC面で描いた絵と同じ配色でバックグラウンド面にコピーするためには、バックグラウンド用のパレットコードを1に指定してください。（CGSET1，1）



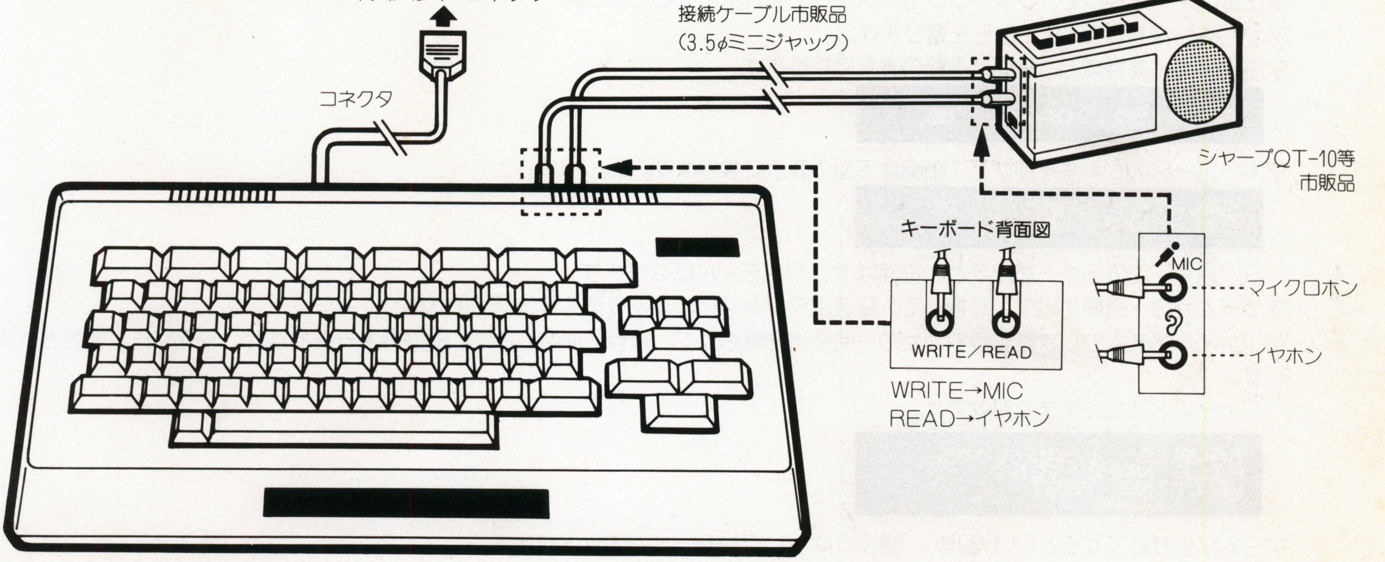
バックグラウンド面にBG　GRAPHIC面で描いた背景画が、コピー（複製）されるので、BG　GRAPHICと同じ絵にスプライト面のマリオが重なって表示されます。

★プログラムの変更や修正をする時の注意

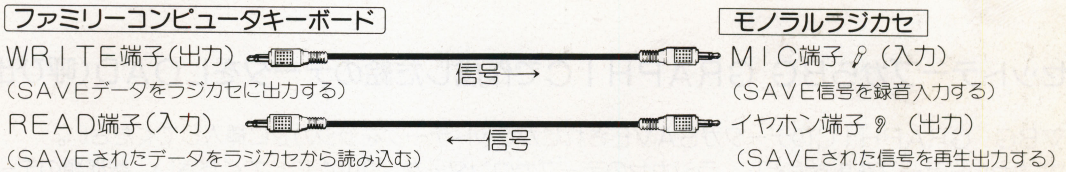
- BASICプログラムの作成や変更および修正は、必ずBG　GRAPHIC(背景)画面を消してから行なってください。
  背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG　GRAPHIC画面を消すにはSHIFTキーを押しながらCLR HOMEキーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

プログラムのSAVE(保存)、LOAD(呼び出し)

BG　GRAPHICで描いた背景の絵のデータやBASICプログラムをカセットテープにSAVE（保存）したり、カセットテープからLOAD（呼び出し）することができます。
操作の方法は、以下の説明に従ってください。



接続ケーブル(3.5φミニジャックが付いているシールド線）のつなぎ方



※接続ケーブルは、ケーブルの両端に3.5φミニジャックがついているシールド線をご使用ください。
抵抗入り接続ケーブルは使用できませんのでご注意ください。

ラジカセのツマミ調整について

SAVE、LOADする場合は、ラジカセのツマミを以下の条件に調整してください。

・モード…………TAPEモードにする
（カセットテープへ録音したり、カセットテープを再生するモード）
・音量調整…………ご使用の機器によって異なりますので、最適な位置を見つけてください。
（通常、ボリュームツマミの中間位置にセットしてください）
・音質調整…………高音（High）にする
（高音が最もよく出る）
・使用テープ………ノーマルテープ
（ノーマルバイアス、ノーマルイコライザー用テープ）

※プログラムは、なるべくカセットテープ１本につき１つだけSAVEしてください。
プログラムをLOADするとき、ファイル名を省略でき、またテープを記録した部分まで合わせる必要がないので便利です。
※QT－10をご使用の場合、クロームポジションテープやメタルテープは使えません。
ラジカセの取扱説明書に従って正しい種類のテープをご使用ください。

※接続ケーブルのラジカセへの 接続やツマミの調整は、モノラルラジカセの取扱説明書をお読みください。

◆FILE(BG GRAPHICの画面)で背景の絵をSAVE(保存)、LOAD(呼び出し)する

★BG GRAPHICで作成した絵をカセットテープへSAVE(保存)する操作手順

1. カセットテープ(ノーマルテープ)をラジカセに挿入してください。
2. 記録したい部分までテープをすすめてください。1本のカセットテープに数種類のデータを記録する場合、テープカウンターを目安にすることができます。
3. [ESC]キーを押して、ファンクションメニューを表示してください。
4. [▼]キーを押して、FILEを選びます。
5. [SPACE]キーを押すと、画面は下図の表示になります。

```
SAVE(S),LOAD(L)?■
```

6. キーボードの[S]キーを押すと、画面は下図の表示になります。

```
SAVE(S),LOAD(L)?S
FILE NAME"■
```

   [S]、[L]キー以外のキーを押すと、画面はセレクトモードになります。
7. ラジカセの一時停止ボタンを押して、録音ボタンを押します。(録音スタンバイの状態)
8. ファイル名を入力します。ラジカセの一時停止を解除して、録音を開始します。[RETURN]キーを押すと、SAVEが開始されます。
   ファイル名は16文字まで有効です。

```
SAVE(S),LOAD(L)?S
FILE NAME"TEST
WRITING TEST
```

9. SAVEが終了すると(1分30秒)、画面にはセレクトモードが表示されます。
10. ラジカセの停止ボタンを押して、録音を停止してください。

★カセットテープからBG GRAPHICで作成した絵のデータをLOAD(呼び出し)する操作手順

1. BG GRAPHICのデータがSAVEされたカセットテープをラジカセに挿入してください。
2. カセットテープをまきもどし、ラジカセのテープカウンターを"000"にしてください。数種類のデータが記録されている場合、先頭以降のデータを探すとき、テープカウンターが目安になります。テープをすすめ、LOADするデータが記録されているテープカウンターに合わせます。
3. [ESC]キーを押して、ファンクションメニューを表示してください。
4. [▼]キーを押して、FILEを選びます。
5. [SPACE]キーを押すと、画面は下図の表示になります。

```
SAVE(S),LOAD(L)?■
```

6. ラジカセの一時停止ボタンを押して、再生ボタンを押します。(再生スタンバイの状態)
7. キーボードの[L]キーを押すと、画面は下図の表示になります。

```
SAVE(S),LOAD(L)?L
FILE NAME"■
```

   [S]、[L]キー以外のキーを押すと、セレクトモードになります。
8. LOADしたいファイル名を入力し、[RETURN]キーを押すと、LOADが開始されます。ラジカセの一時停止を解除して、再生を開始します。
   ファイル名で指定した、BG GRAPHICのデータが見つかるまでスキップ(読み飛ばし)します。
   ファイル名を入力しないで、[RETURN]キーを押すと、テープの先頭のファイルを読み込みます。

```
SAVE(S),LOAD(L)?L
FILE NAME"TEST
SKIP
LOADING TEST
```

   LOAD途中で[STOP]キーを押す、または読み込みエラーが発生した場合は、SELECTになります。
9. LOADが終了すると、BG GRAPHICの画面に読み込んだ絵が表示され、画面はセレクトモードになります。
10. ラジカセの停止ボタンを押して再生を停止してください。

※エラーが発生し、読み込めない場合は、48ページをご覧ください。

※読み込みエラーが発生した場合、または読み込み中、途中で[STOP]キーを押して、LOAD(呼び出し)を中断した場合、BG GRAPHIC面には、未完成な絵が出る場合があります。CLEARするか、GAME BASICモード画面へ戻ってから、再度BG GRAPHICを選択して画面をきれいにしてください。

◆BASICプログラムをSAVE（保存）、LOAD（呼び出し）する

★カセットテープへプログラムをSAVE(保存)する操作手順

▽1 カセットテープ（ノーマルテープ）をラジカセに挿入してください。
▽2 1本のカセットテープに数種類のプログラムを記録する場合、テープカウンターを目安にすることができます。記録したい部分までテープをすすめてください。
▽3 ラジカセの一時停止ボタンを押して、録音ボタンを押します。(録音スタンバイの状態)
▽4 キーボードから、SAVE命令を入力してください。

```
SAVE"TEST"■
```

└── プログラム名は16文字以内でつけてください。

▽5 ラジカセの一時停止を解除して、録音を開始します。
▽6 RETURNキーを押すと、プログラムデータはキーボードからラジカセへ出力されます。画面は下図の表示になります。

```
SAVE"TEST"
WRITING TEST
```

▽7 プログラムのSAVE(プログラムデータの出力)が終了すると、画面は下図の表示になります。

```
SAVE"TEST"
WRITING TEST
OK
■
```

▽8 ラジカセの停止ボタンを押して止めてください。
▽9 正しくSAVEされているかどうかを確かめてください。（LOAD?(照合) する方法をご覧ください）

※カセットテープのリーダーテープ部分が3秒以上の長さの場合には、プログラムが正確にSAVEされないことがあります。長いリーダーテープのカセットテープを使用されるときは、巻き始め部分を、無録音にてテープを少し進めてからSAVEを行なってください。

★カセットテープへプログラムが正しくSAVEされたかどうかを、LOAD?(照合)で確認する操作手順

▽1 SAVEしたカセットテープを、SAVEを開始した所まで巻き戻します。
▽2 キーボードから、LOAD? 命令を入力してください。

```
LOAD?■
```

▽3 RETURNキーを押した後、ラジカセの再生ボタンを押すと、カセットテープが再生されます。画面は下図の表示になります。

```
LOAD?
LOADING TEST
■
```

└ プログラム名

▽4 LOAD? が終了し、プログラムが正しくSAVEされている場合、画面は下図の表示になります。

```
LOAD?
LOADING TEST
OK
■
```

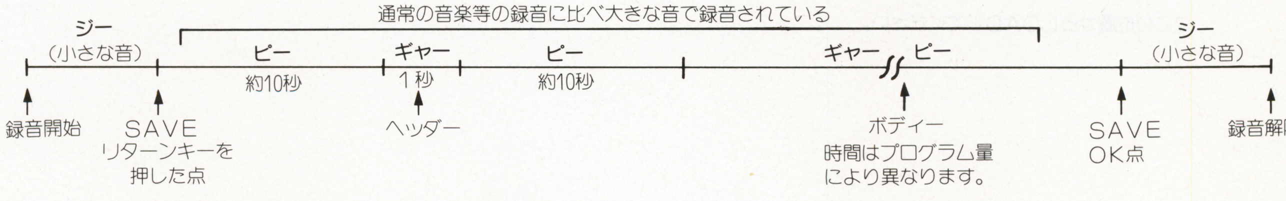
プログラムが正しくSAVEされていない場合、画面は下図の表示になりますので、再度SAVEを行なってください。

```
LOAD?
LOADING TEST
?TP ERROR
OK
■
```

再度、SAVEを行った後でもエラーとなる場合は、カセットテープの録音状態をチェックしたり、カセットテープを交換するなどし、SAVEと、LOAD? を行なってください。

※エラーが発生した場合については、次のページをご覧ください。

プログラムデータが録音されたテープをラジカセのスピーカーから再生すると次の様な音がきこえます。



★カセットテープからプログラムをLOAD(呼び出し)する操作手順

▽プログラムがSAVEされているカセットテープをラジカセに挿入してください。
LOADしたいプログラムの最初の部分まで頭出しをしておいてください。
▽キーボードから、LOAD命令を入力してください。

```
LOAD"TEST"■
```

SAVEされている読み込みたいプログラム名

指定したプログラム名が見つかるまで読みとばします。
▽RETURNキーを押した後、ラジカセの再生ボタンを押すと、プログラムはラジカセからキーボードへ読み込まれます。画面は下図の表示になります。

```
LOAD"TEST"
LOADING TEST
```

※LOAD RETURNキーのみを押した場合は、最初に見つけたプログラムを読み込みます。
▽プログラムのLOADが終了すると、画面は下図の表示になります。

```
LOAD"TEST"
LOADING TEST
OK
■
```

プログラムが正しくLOADされていない場合、画面は下図のエラー表示になりますので、再度LOADを行なってください。

```
LOAD"TEST"
LOADING TEST
?TP ERROR
OK
■
```

※エラーについては、下記をご覧ください。

★LOAD(呼び出し)、LOAD?(照合)時のエラーについて (BG GRAPHICのSAVE、LOADエラーについてもこの項を参考にしてください。)

BASICプログラムのLOAD?やLOADをした際、エラー表示(?TP ERROR)され、ラジカセからプログラムが読み込めない場合は、次のことに注意して、再度SAVE、LOAD、LOAD?を行なってください。

SAVEが正しく行なわれていない場合

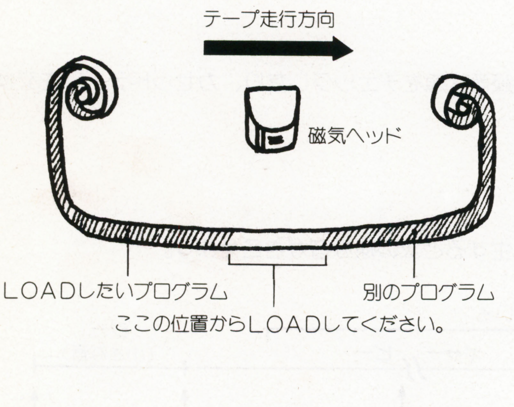
(原因) ●キーボードとモノラルラジカセの接続や、ラジカセの取り扱いが誤っている。
●カセットテープの種類がラジカセに合っていない。
●カセットテープにキズがあったり、テープに録音ムラが生じている。
●SAVE中に外部から雑音が混入した。

(処置) ●SAVEするカセットテープの位置を変えるか別のカセットテープを使用してください。

SAVEは正しいが、LOADが正しく行なわれない場合

(原因) ●キーボードとモノラルラジカセの接続や、ラジカセの再生音量調整等が適当ではない。

(処置) ●ラジカセのツマミを再調整(ボリュームを少しずつ下げてみる、または、音質をかえてみる)したり、接続ケーブルを点検してください。
●SAVEされている位置を確認して、正確に録音の開始点から、カセットテープをLOADしてください。



SAVEされたプログラムの先頭以外(別のプログラムの途中やLOADしたいプログラムの途中など)からLOADを実行しても、正しくLOADされません。必ず、LOADしたいプログラムの先頭からLOADを実行してください。
お手持ちのラジカセをご使用される場合(オーディオ用モノラルラジカセをご使用ください)、機種によっては正しく(SAVE、LOAD)できない場合がありますのでご注意ください。

BASIC 文法編

BASICの仕様水準

文字の種類	数字・英文字・カナ・記号
数の表現範囲 （整　数）	10進（−32768〜＋32767） 16進 （&H0000〜&HFFFF） 文字列（0〜31文字）
変数名の種類	英字で始まる先頭2文字を識別 長さ255文字まで
行番号の範囲	0〜65535
テキスト行の最大桁	255桁
配列の大きさ	2次元まで、要素数制限なし （メモリの範囲内）
マルチステートメント	可能　　：コロンで区切る
サブルーチン、ネスト	制限なし（メモリの範囲内）
FOR　NEXTループ	制限なし（メモリの範囲内）
編集機能	スクリーンエディタ

画面モード	BG　GRAPHIC面、スプライト面、バックグラウンド面 バックドロップ面
グラフィック精度	バックグラウンド面 （28文字×24行） スプライト面（256×240ドット） 1キャラクタ（8×8ドット）
カラー	カラーゼネレータ　52色発生
サウンド機能	音階、テンポ、3重音、音色
コントローラ入力	左・右コントローラ （方向入力、 トリガー入力可能）
ファイル機能	カセットテープ（1,200ボー）
命令数	基本74
アニメキャラクタ設定	既設16種類より選択

※数の表現範囲である−32768〜＋32767を超える数値を扱う演算結果については、保証されません。

ご注意

このマニュアルは、NS-HUBASIC（エヌエス　ヒューベーシック）に基づいて作成されています。

(1)システムソフトウェアおよび本書の内容は、改良のために予告なく変更することがありますので、バージョンナンバー（NS-HUBASIC V1.0）には、特にご注意されるよう、お願いいたします。

(2)本機のシステムソフトウェアならびに本書の内容を当社に無断で複製することは禁止します。

(3)本機は非常に複雑な機能および組合わせを有する製品であり、出荷に際してマニュアルを含め充分なチェックをして万全を期しておりますが、万一ご使用中ご不審な点、お気付きのことがありましたら、もよりの任天堂㈱、各支店・営業所までご連絡ください。
なお、運用した結果、生じる影響については、責任を負いかねますので、あらかじめご了承ください。

機能別命令語索引

〔　　〕内は各命令語の省略形を示しています。

コマンド

一般ステートメント

画面制御ステートメント

MOVEコマンド

特殊ステートメント

サウンド制御ステートメント

関数

●数値関数

●文字関数

●特殊関数

●入出力用文字関数

スプライト制御ステートメント

演算子

演算に使われる+、ーなどの記号のことを演算子といいます。演算子には、算術演算子、関係演算子、および論理演算子の3種類あります。

算術演算子

算術演算子には次の5つがあります。

〔算術演算子〕	〔演算内容〕	〔例〕	〔数学の表現〕
+	加 算	A+B	A+B
ー	減 算	AーB	AーB
＊	乗 算	A＊B	AB
／	除 算	A／B	A÷Bまたは$\frac{A}{B}$(ただし、B≠0)
MOD	剰 余	A MOD B	余り

MODは除算をしたときの余りを計算します。
たとえば、10÷3=3余り1ですから
10MOD3 RETURN
を入力すると、結果は1となります。
演算は次の順位にしたがって行われます。この順位を優先順位といい、優先順位が等しい場合は式の左側にある演算子から先に演算されます。

〔優先順位〕〔算術演算子〕
1.　＊ ／
2.　MOD
3.　+ ー(加減)

演算の順序を変える場合は、先にしたい演算を( )で囲みます。

〔BASICの表現〕	〔数学の表現〕
A＊X+Y	AX+Y
(X+Y)／2	$\frac{X+Y}{2}$
X+Y／2	$X+\frac{Y}{2}$

注)NSーHUBASICは整数型です。小数点を含む演算はできません。
除算の場合の計算結果は、小数点以下を切りすてた値で表示されます。

関係演算子

関係演算子は2つの定数や変数を比較するときに用います。結果は、真のときー1、偽のとき0が得られます。
下に関係演算子を示します。

〔関係演算子〕	〔意味〕
=	両辺が等しい
<>	両辺が等しくない(≠)　><の書きかたは使用できません。
>	左辺が右辺より大きい
<	左辺が右辺より小さい
>=	左辺が右辺より大きいか等しい(≧)　=>の書きかたは使用できません。
<=	左辺が右辺より小さいか等しい(≦)　=<の書きかたは使用できません。

関係演算子は、プログラムの内のIF文の中では次のように使います。
IF X>0 THEN 1000 〔意味〕Xの値が0より大きければ1000番にジャンプしなさい。
詳しくは、IFーTHEN文を参照して下さい。
また、
A=X>0 またはA=(X>0)
のように使うと、Xが0より大きければAにはー1が代入され、Xが0より小さいか等しければAには0が代入されます。したがってAの値からXが0より大きいか否かを知ることができます。

論理演算子

論理演算子はビット単位で∅と１の間のブール演算やその他のビット操作（これを論理演算といいます）を行う記号で、次のように４つあります。

〔演算順位〕	〔論理演算子〕	〔省略形〕	〔意味〕	
1.	NOT	NO.	not	否 定
2.	AND	A.	and	論理積
3.	OR	ありません	inclusive or	論理和
4.	XOR	XO.	exclusive or	排他的論理和

論理演算とは、真を１、偽を∅とする２つの値の間で行われ、結果（真理値）も１（真）か∅（偽）の２つの値で表されます。次に上の６つの論理演算子のそれぞれについて、具体的な演算操作を真理値表としてまとめます。

NOT（否定）

X	NOT X
1	∅
∅	1

注）これはXが１のときNOTXは∅になりXが∅のときNOTXが１になることを示しています。

AND（論理積）

XY	X AND Y
11	1
1∅	∅
∅1	∅
∅∅	∅

注）これはXが１かつYが１のときのみX AND Yが１になり、それ以外は∅になることを示しています。ちょうどXとYの乗算とその積に対応しています。

OR（論理和）

XY	X OR Y
11	1
1∅	1
∅1	1
∅∅	∅

注）これはXとYの加算とその和に対応しています。

XOR（排他的論理和）

XY	X XOR Y
11	∅
1∅	1
∅1	1
∅∅	∅

注）排他的というのはXとYとが同じ値のときに∅（＝偽）になると定めたからです。

注）変数と論理演算子をつづけて書く場合は、間にスペースをあけてください。

```
(例)●NOT 13    13=(0000 0000 0000 1101)₂
                ∴NOT13=(1111 1111 1111 0010)₂
                       =-14
    ●15AND5    15=(0000 0000 0000 1111)₂
                 5=(0000 0000 0000 0101)₂
                ∴15AND5=(0000 0000 0000 0101)₂
                       =5
    ●50 OR 44  50=(0000 0000 0011 0010)₂
               44=(0000 0000 0010 1100)₂
                ∴50 OR 44=(0000 0000 0011 1110)₂
                       =62
    ●42XOR36   42=(0000 0000 0010 1010)₂
               36=(0000 0000 0010 0100)₂
                ∴42XOR36=(0000 0000 0000 1110)₂
                       =14
```

※ $(000\cdots)_2$→２進数であることを示します。

論理演算子をIF文の中で用いてプログラムの流れを変えることができます。

（例） IF X>∅ AND X<1∅ THEN 1∅∅∅

〔意味〕Xの値がX>∅かつX<10ならば1000番にジャンプしなさい。

IF X<∅ OR X>1∅ THEN 1∅∅∅

〔意味〕Xの値がX<∅またはX>10ならば1000番にジャンプしなさい。

３種類の演算子には、次のような優先順位がつけられています。

1．算術演算子
2．関係演算子
3．論理演算子

このことから演算の優先順位をまとめると次のようになります。

1．（ ）で囲まれた部分
2．関数
3．＊、／（乗除）
4．MOD（剰余）
5．＋、－（加減）
6．=、<>、>、<、>=、<=、
7．NOT（否定）
8．AND（論理積）
9．OR （論理和）
10．XOR（排他的論理和）

特殊な記号

BASICでは、演算子のほかにも特別な意味をもつ、次に示すような記号があります。

(1)−(ハイフン)

LIST文など行の範囲を指定するとき、どこからどこまでの行かという場合に使います。

(例) LIST 100-300

(2) ,(カンマ)

PRINT文、INPUT文、DATA文などオペランドが並ぶ場合、その区切り記号（セパレータ）として使用されます。

(例)
```
PRINT A,B,C
INPUT A,B,C
DATA 10,13,9,14,13
```

(3) :(コロン)

マルチステートメントの区切り記号として使います。

(例)
```
A=B-C:PRINT A
FOR I=0 TO 9:PRINT X$(I):NEXT
X1=X:Y1=Y:GOSUB10000:GOSUB20000
```

(4) ;(セミコロン)

PRINT文などの区切り記号として使います。

(例)
```
PRINT"コタエ=";A
INPUT"A=";A
```

(5)?(クエスチョンマーク)

PRINT文の代用として使えます。

(例) ?A,B,C

(6)&H○○は16進数を示します。

それ以外は10進数です。

(例) &H2Cは、10進数で44のことです。

文法編のみかた

この章では、NS−HUBASICの命令語のすべてを解説します。なお、命令語は、次のような構成で解説しています。

| 働　　き | 命令語の働きを説明したものです。これを読むことによって命令語の意味をつかむことができます。 |

| 文　　法 | 命令語の文法が一般的な形で書かれてあり、使い方を知ることができます。 |

なお、ここで使っている記号や文字には次のような一定の約束があります。

1.英字の大文字はそのまま入力します。
2.〔　〕の中は、ユーザーが任意に省略することができます。
3.｛　｝の中に縦に並べて列記してある項目はユーザーが任意に選択します。
4.(　)、「，」(カンマ)、「：」(コロン)、「；」(セミコロン)、「−」(ハイフン)、「＝」(符号)などの記号は指定された位置に正しく入力します。
5.……は、1行(255文字)の範囲で任意の回数繰り返すことができることを意味します。
6.「　」(空白)のあるところは空白をとることを意味しますが、この空白を無視してつめてよい場合もあります。
7.〝　〞の中は文字列データであることを示しています。
8.そのほか文法の中で使っている語句については文法のすぐ下の部分か 説　明 のところで詳しく説明しています。

| 省 略 形 | 命令語の短縮形を示します。 |

| 説　　明 | 命令の使用方法や働き、注意事項を詳しく述べてあります。 |

☞「参照してください」の意味で、関連した命令語や項目が示されています。

| サンプル プログラム | 適当な例題やその実行結果等を示してあります。 |

プログラムリストとして書かれてあるため、画面表示とは異なる場合があります。同じ行番号のプログラムは画面上、行が変わっても連続して入力してください。

デフォルト値……命令語のパラメータを指定しないとき、パラメータの値として自動的にセットされる値のことです。

GAME　BASICモード画面からBASICの面へ入った時は、つぎの設定となっています。

CGEN2
SPRITE　OFF
POSITION n，120，120
PLAY〝O3V15T4M0〞及び音程の長さ5

カラーパレットは、バックグラウンド用、スプライト用共にバックグラウンド用のパレットコード1の配色となっています。

コマンド

CLEAR

働き　BASICが使用するメモリ領域の最上位アドレスを指定します。

文法　CLEAR アドレス
アドレス→BASICで使用するメモリ領域の最上位アドレス
&H77FF以下
（104ページのメモリマップ参照）

省略形　CLE.

説明　メモリ内でBASICが使用する領域の最上位アドレスを指定するためのものです。
マシン語プログラムをBASICプログラムと併用する場合BASICのプログラムや変数によって破壊されないように、BASICが使用するメモリ領域の最上位アドレスを指定する必要があります。
CLEARで最上位アドレスを指定すると同時にFOR－NEXT、GOSUBなどのネスティング及び、変数、配列はクリアされます。
このコマンドは通常プログラムの先頭部分に書かれ、サブルーチンの中では使用できません。また、一度指定したアドレスは再びCLEARで指定されるまで有効となります。
CLEARでアドレスを省略すると、すべての変数と配列変数について、数値変数は0に、文字変数は"　"（ヌルストリング）にクリアします。
☞61ページCLEAR参照

サンプルプログラム

```
10 REM * CLEAR *
20 CLEAR &H7600
```

BASICで使用できるメモリ領域の最上位アドレスを&H7600に指定しています。
PRINT　FREで確認してください。

NEW

働き　プログラムをすべて消します。

文法　NEW

省略形　ありません。

説明　新しいプログラムを入れる場合には、その前にそれまでメモリに入っていた古いプログラムを消しておくとトラブルが起きません。そこで、新しいプログラムを入れる場合にはNEWというコマンドを入れて、古いプログラムを消します。
NEWはプログラムを消すだけでなく、メモリに入っている変数の内容もすべて消します。ただし、メモリのユーザエリアと画面の状態は、NEWを実行しても変わりません。
なお、誤ってプログラムを消してしまうことのないように、NEWには省略形がありません。使用には十分注意してください。
（プログラム上では使用しないでください）

サンプルプログラム

```
10 REM * NEW *        } プログラムを入力する。
20 X=999
30 PRINT X
LIST                  ……プログラムのリストをみる。
10 REM * NEW *        } プログラムが誤りなく入力されている。
20 X=999
30 PRINT X
OK
RUN                   ……入力されたプログラムを実行してみる。
 999                  ……Xの値999が画面表示される。
OK
NEW                   ……プログラムを消す。
OK
LIST                  ……プログラムのリストをみる。
OK                    ……プログラムは消えている。
■
```

LIST

働　き　メモリ中のプログラムを画面に表示します。

文　法　LIST〔〔m〕〔－〔n〕〕

m→表示させる最初の行番号

n→表示する最後の行番号

省略形　L.

説　明　LISTは、メモリに入っているプログラムを画面上に出力させるコマンドで、メモリ中のプログラムを調べたり、修正したりするときに使います。行番号を指定することによって、プログラムの任意の部分を自由に表示することができます。

LISTm……………………………m番のみを表示…(1)

LISTm，またはLISTm－……m番以降を表示…(2)

LISTm，nまたはLISTm-n …m番からn番を表示……………(3)

LIST，nまたはLIST -n……n番までを表示…(4)

LIST ……………………………全体を表示………(5)

指定した行番号がプログラム中になかった場合、BASICは次のように表示します。

(1)のときは、何も表示しないですぐ次の入力待ちの状態になります。

(2)のときは、mより大きくて1番近い番号から出力を始めます。m番以降がなければ、何もしないですぐに次の入力待ちになります。

(3)のときは、初めのmについては(2)と同じ動作を、後ろのnについては、(4)と同じ動作をして、表示を終わります。

(4)のときは、最初からnより小さくて1番近い番号まで表示します。n番以降がなければ、何もしないですぐ次の入力待ちになります。

なお、ESCを押すと表示を一時的に停止することができます。表示を再開するときは任意のキーを押します。

サンプルプログラム

```
10 REM * LIST *
20 INPUT X
30 Y=X*X
40 PRINT X;Y
50 END
```
プログラムを入力する。

```
LIST 30
30 Y=X*X
OK
```
(1)の例、行番号30のプログラムだけをリストする。

```
LIST 40-
40 PRINT X;Y
50 END
OK
```
(2)の例、行番号40以後のプログラムをリストする。

```
LIST 20-40
20 INPUT X
30 Y=X*X
40 PRINT X;Y
OK
```
(3)の例、行番号20から40の間にあるプログラムをリストする。

```
LIST -20
10 REM * LIST *
20 INPUT X
OK
```
(4)の例、行番号20までのプログラムをリストする。

```
LIST
10 REM * LIST *
20 INPUT X
30 Y=X*X
40 PRINT X;Y
50 END
OK
■
```
(5)の例、すべてのプログラムをリストする。

RUN

働　き　プログラムを実行させます。

文　法　RUN n

n→実行を開始させたい行の行番号

省略形　R.

説　明　RUNは、プログラムを実行させるためのコマンドです。RUNを入力すると、プログラムは最初から順に実行されますが、このとき変数はすべてクリアされます。

また、プログラムを途中から実行させたい場合は、実行開始行番号nをつけます。

(例　RUN　1000　RETURN)

STOPを押してプログラムを止めたあと、再度実行するときに、RUNやRUNnとすると、それまでの変数に入っていた内容をすべてクリアしてしまうので、変数をクリアせずに実行したい場合は、GOTO(GOTO文参照)、またはCONTを使用してください。

(例　GOTO　1000　RETURN)

☞　GOTO、CONT参照

サンプルプログラム

```
10 REM * RUN *
20 PRINT "ﾊｲ!"
30 PRINT "ｺﾝﾆﾁﾊ"
40 END
```
プログラムを入力する。

```
RUN
```
……プログラムを実行する。

```
ﾊｲ!
ｺﾝﾆﾁﾊ
```
プログラムが実行される。

```
OK
■
```

CONT

働き　プログラムを再開します。

文法　CONT

省略形　C.

説明　プログラム上のSTOP命令によってプログラムが停止している場合や STOP キーを押してプログラムが停止している場合に、CONTを入力すると、プログラムは止まった次の行番号の命令文から実行を開始します。この場合は、停止する前までの変数は保持されたままになっています。
ただし、ENDやエラーの発生によってプログラムが止まったときや、CLEAR（CLEAR文参照）実行直後、あるいはプログラムを止めてからプログラムの書き換えを行ったときには、CONTは正しく実行されず、エラー（CC ERROR）が出ます。なお実行が再開できるときはOK.（ピリオドつき）が表示され、再開できないときはOK（ピリオドなし）が表示されます。

☞ STOP、END参照

サンプルプログラム

☞STOP参照

LOAD

働き　カセットテープからプログラムを読み込みます。（⇔ SAVE）

文法　LOAD〔"ファイル名〕
　ファイル名→プログラムについている名前　16文字以内

省略形　LO.

説明　カセットテープに記録されているプログラムを読み込んでメモリに記録するには、LOADコマンドを使います。
ファイル名を指定すると、その名前のプログラムが見つかるまでほかのプログラムを読み飛ばし、指定されたプログラムを読み込みます。ファイル名は、カセットテープの最初に現れたプログラムを読み込む場合に限り省略します。
使用方法は45ページをご覧ください。

サンプルプログラム

```
LOAD
LOADING TEST
OK
■
```

……ファイルネームを指定しないで、プログラムをカセットテープからLOAD（読み込み）します。

```
LOAD"TEST-2"
SKIP TEST
SKIP TEST-1
LOADING TEST-2
OK
■
```

……ファイルネームを指定してLOADする。

……ファイルネームが一致しないプログラムは、SKIP（読み飛ばし）します。

SAVE

働き　プログラムを カセットテープに記録します。（⇔ LOAD）

文法　SAVE〔"ファイル名"〕
　ファイル名→記録するプログラムにつける名前　16文字以内

省略形　SA.

説明　記憶されているプログラムをカセットテープに記録するには、SAVEコマンドを使います。
ファイル名は、記録するプログラムにつける名前で「"」（ダブルクォーテーション）で囲んで使います。
ファイル名は省略することができますが、いくつかのプログラムを区別したり、整理するときには、ファイル名をつけておいたほうが便利です。
使用方法は 45 ページをご覧ください。

サンプルプログラム

```
SAVE
WRITING
OK
■
```

……ファイルネームをつけないでSAVEします。

```
SAVE"TEST"
WRITING TEST
OK
■
```

……ファイルネーム（例　TEST）をつけてプログラムを カセットテープにSAVE（保存）します。

LOAD?

働き	SAVEしたプログラムが、ファイルに正しく記録されたかを照合します。
文法	**LOAD?〔"ファイル名"〕**
省略形	**LO.?またはLO.P.**
説明	カセットテープに記録されているプログラムがメモリに記録されているプログラムと同じものかどうかを照合する場合にLOAD?コマンドを使います。 ファイル名（プログラムについている名前。16文字以内）を指定すると、その名前のプログラムが見つかるまで、他のプログラムを読み飛ばし、指定のプログラムがみつかれば、照合を始めます。 ファイル名を省略すると、最初に見つけたファイル名のプログラムとメモリにあるプログラムとを照合します。 カセットテープにプログラムをSAVEした場合、LOAD?で正しく記録されたかを調べることが大切です。 使用方法は47ページをご覧ください。

サンプルプログラム

```
LOAD?                  ……ファイルネームのない場合。
LOADING TEST           ……照合中。
OK                     ……照合結果OK照合完了。
■

LOAD?"TEST"            ……ファイルネームをつけた場合。
LOADING TEST
OK
■

?TP ERROR  ┐
OK         ┘照合結果不可の場合。
■
               再度、SAVEしなおしてください。
```

一般ステートメント

代　入

命令語はありません。

働　き　変数に式の値を代入します。

文　法　**変数＝式**

説　明　代入は、「＝」で結ばれた右辺の式の値を、左辺の変数値に代入します。
右辺の式には単なる定数や変数も含まれ、左辺には必ず変数のみがかかれます。右辺の変数は、その前（直前とは限らない）の行で定義されている必要があります。もし定義されていなければ、変数は数値変数のときに0、文字変数のときに〝　〞（ヌルストリング）とみなされます。
また、「＝」で結ばれる両辺の型は、左辺が数値変数のとき右辺は数値型の式、左辺が文字変数のとき右辺は文字型の式というように、一致していなければなりません。（文字変数と数値変数の混用はできません）

（正）　X＝A　Y＝A＊X＋10　　X＄＝〝＆H〞＋A＄
（誤）　X＄＝A　Y＄＝A＊X＋〝10〞
　　　　X＝〝＆H〞＋A＄

サンプルプログラム

```
10 REM ---ダイニュウ---
20 A=10              ……数値変数Aに数値10を代入します。
30 A$="ABC"          ……文字変数A＄に文字ABCを代入します。
40 PRINT A$;A        ……変数A＄、Aをプリントします。
RUN
ABC 10               ……A＄、Aに代入されています。
```

PRINT

働　き　画面に計算結果などの情報を表示します。

文　法　PRINT ［定数／変数／式］［；／，［定数／変数／式］…………］

定数、変数、式→画面に表示される情報

省略形　？またはP．

説　明　PRINTは、画面上に定数、数値変数の値、文字変数の値（文字列）、および式の値（演算結果）を出力します。
PRINTは、「印刷する」という意味ですが、これは初期の出力装置のほとんどがテレタイプであったことに由来しています。
PRINTはよく使われるステートメントなので、省略形として「？」が用意されています。
PRINTによって出力される値は、「；」か「，」で区切ることによって複数個続けて書くことができます。このときの「；」や「，」を区切り記号（セパレータ）といいます。
区切り記号として「；」を使うと、文字をすぐ後に続けて書くことができます。ただし、数値の前の符号の部分（正の数のとき）と後ろには必ず「　」（ブランク）が1つずつとられます。
区切り記号として「，」を使うと、8文字単位で画面上に続けて出力します。画面の表示領域は次のように4つのブロックに分割されていて、出力情報は必ず各ブロックの頭から表示されます。

画面表示例

```
|←──────────── 表示領域 ────────────→|

    8文字        8文字        8文字      4文字
|-----------*-----------*-----------*--------|
0          7 8        15 16       24 25     28
例①A          B           C           D          PRINT "A", "B", "C","D", "E", "F"
  E          F
②ABC        ABC         ABC         ABC        PRINT "ABC", "ABC", "ABC", "ABC"
③ABCDEFG    ABCDEFG                             PRINT "ABCDEFG", "ABCDEFG"
④ABCDEFGHI              ABCDEFGHI               PRINT "ABCDEFGHI", "ABCDEFGHI"
```

なお、文字定数を書く場合は、後ろに区切り記号がないときに限って、後ろにつける「〞」を省くことができます。また、PRINT文の最後に「；」をつけると、次のPRINT文の内容が接続して表示されます。
PRINTを単独につかうと、1行改行が行われます。

ＩＮＰＵＴ

働　き　キーボードから数値や文字を入力します。

文　法　ＩＮＰＵＴ〔"文字列"〕{；/，}変数〔，変数，…〕

文字列→画面に表示する文字列
変数→キーボードから入力する数値や文字が入る変数

省略形　Ｉ.

説　明　キーボードから入力した数値や文字のデータを変数に入れます。あるデータを入力する場合には、前もってデータを入れる変数を用意しなければなりません。そのため、ＩＮＰＵＴのあとには、必ず変数をつけます。変数は数値変数、文字変数どちらでもつけることができます。また、変数は「，」（カンマ）で区切ることにより複数個つけることができます。この場合はデータを入力するとき、対応するデータを「，」で区切って１度に入力します。入力するデータの数は、ＩＮＰＵＴのあとにつけた変数の数と一致していなければなりません。
文字変数には、ほとんどすべての文字を入れることができますが、「，」(カンマ)は「"」(ダブルクォーテーションマーク)で囲まなければ入力できません。これは「，」が区切り記号として使われているためです。
ＩＮＰＵＴ文を実行すると、画面に「？」を出して入力待ちの状態となりますが、「？」だけでは何を入力すればよいのかわかりません。そこで、ＩＮＰＵＴには文字列のＰＲＩＮＴ機能がついていて、画面に表示したい文字列を「"」で囲んでから「；」(セミコロン)をはさんで変数を書きます。
また、「？」を出さないようにしたい場合は「"」で囲んだ文字列と変数の間に「，」をはさみます。
文字変数の入力を指定しているときに数値を入れると、数字を文字として受けつけますが、数値変数の入力を指定しているときに文字を入れると変数扱いとなり０が入ります。数値変数、文字変数いずれの場合も、何も入力しないで RETURN キーを押すと現在のその変数のもっていた値や文字は消えます。

サンプルプログラム

次のサンプルは、20と30番で２つのデータを読み込んで、加減乗をするプログラムです。

```
10 REM * INPUT *
20 INPUT"A=";A
30 INPUT"B=";B
40 C=A+B
50 D=A-B
60 E=A*B
70 PRINT"A+B=";C
80 PRINT"A-B=";D
90 PRINT"A*B=";E
```

（コンピュータの扱える数値の範囲は−32768〜＋32767です。）

ＬＩＮＰＵＴ

働　き　キーボードから文字を入力します。

文　法　ＬＩＮＰＵＴ〔"文字列"〕{；/，}文字変数

文字変数→キーボードから入力する文字列が入る変数

省略形　ＬＩＮ.

説　明　ＬＩＮＰＵＴは、ＩＮＰＵＴと同様キーボードからデータを入力するためのステートメントですが、ＩＮＰＵＴと違うのは「，」(カンマ)もデータとして入力できる点です。
また、このステートメントで入力できるのは文字列のみで31文字までで、入力するための変数は１つしか指定できません。
ＬＩＮＰＵＴには、ＩＮＰＵＴと同様文字列のＰＲＩＮＴ機能がついていて、画面上に入力すべき文字についてのメッセージを表示することができます。ただし、ＬＩＮＰＵＴでは、変数にこのメッセージまでデータとして入力されてしまいますので、入力した文字だけデータにしたいときには、プログラム中にメッセージの部分を取り除く処理を入れてやる必要があります。
ただし、この場合入力できる文字数は、メッセージの文字数を含めて31文字以内です。
なお、ＩＮＰＵＴ文のように「？」は出ず、「"」で囲んだ文字列と文字定数の間に「；」と「，」のどちらをはさんでも同じ動作をします。

サンプルプログラム

```
LINPUT"STRING=";A$
STRING=23.5,""",,,;
OK
PRINT A$
STRING=23.5,""",,,;
OK
■
```

CLEAR

働き	変数および配列をすべてクリアします。
文法	CLEAR
省略形	CLE.
説明	メモリ上のすべての変数および配列をすべてクリアします。すなわち、数値型のものを0に、文字型のものをヌルストリング（何も入っていない状態）にします。

サンプルプログラム

```
X$="クリア"       ……X$に文字クリアを代入。
OK
Y=125             ……Yに数値125を代入。
OK
PRINT X$;Y        ……X$とYの値をプリントして確認します。
クリア 125        ……代入されています。
OK
CLEAR             ……クリアします。
OK
PRINT X$     ┐
             │
 OK          │
 PRINT Y     │…クリアされています。
  0          │
 OK          ┘
 ■
```

DIM

働き　配列宣言をします。

文法　DIM 配列名（m1〔，m2〕）〔，配列名（n1〔，n2〕）
，……〕

配列名→配列の名前、数値変数、文字変数いずれも可

省略形　DI.

説明　配列の名前と次元数、および添字のサイズを設定します。
1つのDIM文で複数の配列を宣言できて、配列ごとにメモリの範囲内で2次元までの添字の指定ができます。
配列宣言後、数値型配列には0、文字型配列にはヌルストリングが入り、すべての配列の中味はクリアされます。

サンプルプログラム

(1)数値型配列の例

```
10 REM * DIM --- (1) *
20 DIM A(3),B(3,3)          …配列宣言
30 FOR I=0 TO 3
40 A(I)=I
50 PRINT A(I);
60 NEXT
70 PRINT:PRINT
80 FOR I=0 TO 3
90 FOR J=0 TO 3
100 B(I,J)=I*10+J
110 PRINT B(I,J);
120 NEXT
130 NEXT
```

30〜60：1次元配列A（I）内に数値を代入し表示します。

80〜130：2次元配列B（I，J）内に数値を代入し表示します。

(2)文字型配列の宣言

```
10 REM * DIM --- (2) *
20 DIM A$(3),B$(3,3)          …配列宣言
30 FOR I=0 TO 3
40 A$(I)="TEST"+STR$(I)
50 PRINT A$(I)
60 NEXT
70 PRINT
80 FOR I=0 TO 3
90 FOR J=0 TO 3
100 B$(I,J)="TOKYO"+STR$(I)+STR$(J)
110 PRINT B$(I,J)
120 NEXT
130 NEXT
```

30〜60：1次元配列A$（I）内に文字を代入し表示します。

80〜130：2次元配列B$（I，J）内に文字を代入、表示します。

GOTO

働き	指定した行番号へ無条件にジャンプします。
文法	GOTO〔行番号〕 行番号→ジャンプ先の行番号
省略形	G.
説明	GOTOは指定した行番号に、無条件にジャンプするステートメントで、プログラムの流れを変えるときに使用します。 ダイレクト命令では、定数の値をクリアせずに、特定行番号からプログラムを実行させることができます。 （例　GOTO　2000 RETURN ） ☞RUN参照

サンプルプログラム

```
10 REM * GOTO *
20 A$=INKEY$(0)
30 PRINT A$,ASC(A$)
40 IF A$="Z" THEN END
50 GOTO 20
```

- 20：何かのキーが押されるのを待ちます。
- 30：押されたキーの文字、記号と、そのアスキーコードを表示します。
- 40：Zキーが押されると、プログラムを終了します。
- 50：行番号20へ飛び、行番号20を実行します。

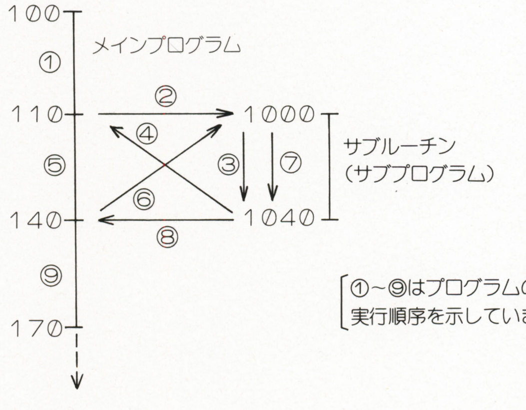
GOSUB

働き	プログラム内のサブルーチンを呼び出します。
文法	GOSUB〔行番号〕 行番号→サブルーチンの開始行番号
省略形	GOS.
説明	IF文（IF−THEN参照）やFOR−NEXT文(FOR−NEXT参照）で作ったループは、同じ処理を何回も続けて行うものでしたが、プログラムの所々で同じ処理を行う必要がある場合には、何回も繰返して使う部分をサブプログラムとして組んでおき、必要に応じてそれをプログラム本体（メインプログラム）から呼び出すようにします。メインプログラムから呼び出されるサブプログラムのことをサブルーチンといい、このサブルーチンを呼び出すステートメントとしてGOSUBが用意されています。 GOSUBのあとについている行番号がサブルーチンの開始行で、呼び出されるサブルーチンの最後には必ずRETURNをつけます。サブルーチンの中にさらに別のGOSUB文を書いてもかまいません。（ただし、メモリの範囲で制限されます）

サンプルプログラム

```
10 REM * GOSUB *
100 FOR I=1 TO 25
110 GOSUB 1000
120 NEXT
130 FOR I=25 TO 1 STEP -1
140 GOSUB 1000
150 NEXT
160 PRINT "END"
170 END
1000 FOR J=0 TO I
1010 PRINT"*";
1020 NEXT
1030 PRINT
1040 RETURN
```

- 100〜170：メインプログラム
- 1000〜1040：サブプログラム



［①〜⑨はプログラムの実行順序を示しています。］

行番号100〜170がメインプログラムです。
行番号1000〜1040がサブプログラムです。
メインプログラムの行番号110と140でサブルーチンを呼び出しています。プログラムの流れは左図のようになっています。

［GOSUB文のサブルーチン内に注釈をつける場合は「REM」は使用できません。「'」を使用してください。］

RETURN

働き	サブルーチンからの復帰をします。
文法	RETURN〔{行番号}〕 行番号→サブルーチンからの戻り先行番号
省略形	RE.
説明	RETURNは、GOSUBで呼ばれたサブルーチンの最後につけて、戻り先に復帰するときに使います。 行番号を指定するとその行に戻りますが、省略するとGOSUB文の後ろに戻ります。

サンプルプログラム

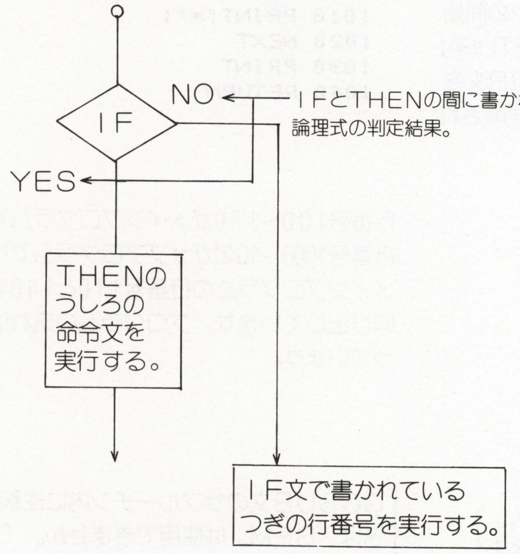
☞GOSUB参照。

IF-THEN

働き	論理式による分岐を行います。
文法	IF 式 THEN {行番号 / 文} 式→論理式　52ページをご覧ください。 行番号→ジャンプ先の行番号 文→任意のステートメント
省略形	IF-T.
説明	IF文は、条件分岐を行うステートメントで、THENと対にして使います。IFとTHENの間に論理式を書き、論理式が成立した場合THEN以降を実行し、成立しなければ、次の行を実行します。たとえば、

IF X=10 THEN 500
また、
IF X=10 GOTO 500
（もしXが10ならば、行番号500へ飛び、行番号500を実行する。）

のように、THENのあとにGOTO文がくる場合、THENを省略することができます。
IF文の中に別のIF文を入れてもかまいません。



サンプルプログラム

```
10 REM * IF - THEN *
20 PRINT "PUSH Y!";
30 A$=INKEY$(0)
40 IF A$<>"Y" THEN BEEP:GOTO 30
50 PRINT:PRINT"Y ガ オサレマシタ."
60 PRINT
70 GOTO 20
```

行番号30でキーから1文字入力を待ちます。
何かキーが押されたら、行番号40で「Y」か否かを判断します。「Y」でなかったら「ピッ」と音を出して行番号30へもどり1文字のキー入力を待ちます。

「Y」のキーが押された場合は、行番号40の論理式（A$<>"Y"）が成立しないため、THEN以後の命令は実行せず、行番号50を実行します。

FOR〜TO〜STEP　NEXT

働き　FOR−NEXTループの間の処理を繰り返し実行します。

文法　FOR　i＝ℓ　TO　m　〔STEPs〕
　　　〜
　　　NEXT

- i→ループ変数
- ℓ→初期値
- m→終了値
- s→増分　負の数でもよい
  デフォルト値は1。STEPsを省略した場合は、STEP1として実行されます。

（ℓ、m、s は）整数で−32768〜＋32767の範囲

省略形　F.-TO-ST.
　　　　〜
　　　　N.

説明　FOR−NEXTループにおいて、FOR文はループの先頭、NEXT文はループの終わりを示しています。たとえば、

```
      ↓ プログラムの流れ
 ┌→FOR　I=1　TO　10　STEP2
 │   ┌──────────┐
 │   │処　　　　理│
 │   └──────────┘
 └─NEXT
      ↓
```

というFOR−NEXTループは次のような動作をします。プログラムの流れがFOR文に来ると、ループ変数Ｉの値が１になって、１回めの処理に入ります。ループの１回めの処理が終わってNEXTを見つけると、Ｉの値に増分２（STEP2による）を加え、Ｉは次の値３となってFORのところまで戻り、２回目の処理に入ります。このようにしてＩの値が３、５、７、９、11と終了値10を超えたところでループを終了し、次の行へ移ります。（Ｉの値としては終了値に等しいか、または、その直前の値になる回数分だけ処理を実行します）

なお、STEPsを省略すると増分は１になります。
NEXT文に対応するFOR文がない場合は、「NF　ERROR」のエラーメッセージが出ます。
ループ変数ｉの初期値ℓと終了値ｍの条件がすでに満たされている場合は、そのままNEXT文の次の文へ移ります。

```
          ↓
 ┌──→FOR　I=1　TO　10
 │        〜
 │ ┌─→FOR　J=1　TO　20
 │ │      〜
 │ │ ┌→FOR　K=1　TO　5
 │ │ │    〜
 │ │ └─NEXT
 │ └───NEXT
 └─────NEXT
          ↓
```

NEXTの後ろには、ループ変数名はつけることはできません。（エラーとなります）

```
     ┌NEXT　K
誤り ┤NEXT　J
     └NEXT　I
```

サンプルプログラム

```
10 REM * FOR-NEXT (1) *
20 FOR I=0 TO 10 STEP 2
30 PRINT I;
40 NEXT
RUN
 0 2 4 6 8 10
OK
■
```

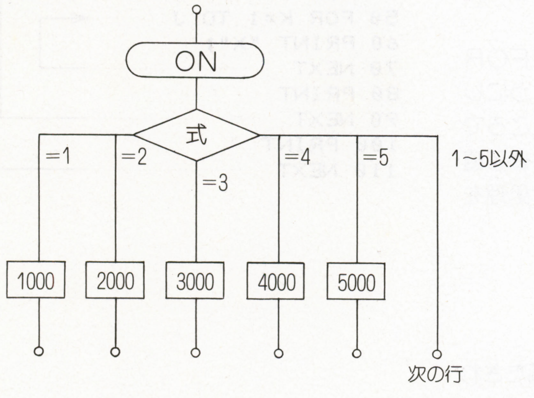
```
10 REM * FOR - NEXT (2) *
20 CLS
30 FOR I=1 TO 20 ←────────┐
40 FOR J=1 TO I ←─────┐   │
50 FOR K=1 TO J ←──┐  │   │
60 PRINT "X";      │  │   │
70 NEXT ───────────┘  │   │
80 PRINT              │   │
90 NEXT ──────────────┘   │
100 PRINT                 │
110 NEXT ─────────────────┘
```

ON

働き	式の値によって、指定された行へジャンプします。
文法	ON 式 { GOTO / GOSUB / RETURN / RESTORE } 行番号〔，行番号，行番号，……〕 式→数値変数型で値が1から始まる整数であること 行番号→0～65534
省略形	O.
説明	式の値が1のとき、{ } の後にかかれた行番号の並びの1番目にある行番号の行にジャンプします。 式の値が1ならば1番目の行番号、2ならば2番目、3ならば3番目、…というように対応しており、式の値が0のとき、または指定した行番号の個数を越えるときは、ON文の次の文に移ります。

```
IF  X=1  THEN 1000
IF  X=2  THEN 2000
IF  X=3  THEN 3000
IF  X=4  THEN 4000
IF  X=5  THEN 5000
            ↓
ON  X  GOTO 1000,2000,3000,
              4000,5000
```

〈フローチャート〉



☞ GOTO、GOSUB、RETURN、RESTORE 参照

サンプルプログラム

```
10 REM * ON-GOSUB *
20 INPUT "1 カラ 6 デ ナンバンガ スキ ";N
30 ON N GOSUB 100,200,300,400,500,600
40 IF N<1 OR N>6 THEN 20
50 PRINT N;"ハ 「";X$;"」 ノ ショウチョウ デス。"
60 GOTO 20
100 X$="エイエン":RETURN
200 X$="キボウ":RETURN
300 X$="ジョセイ":RETURN
400 X$="ダンセイ":RETURN
500 X$="カンゼン":RETURN
600 X$="ケッコン":RETURN
```

行番号20番で選ばれた数値を、行番号30番で判断します。それぞれの数値に対応する行番号(100～600)へジャンプし、各文字変数が与えられ、行番号50番で表示されます。1～6以外の数値を入力した場合、行番号20番目へもどり、再度聞き直してきます。

STOP

働き	プログラムの実行を止めます。
文法	STOP
省略形	STO.
説明	プログラム中にSTOP文が入れてあると、「BREAK IN 行番号」OK. を画面に表示して止まります。このとき、行番号はSTOPの入れてある行を示しています。 STOP文によって止めたプログラムは、変数の中をクリアせず、CONT文（CONT参照）によって次の文から再開できますので、プログラムのデバッグ(虫取り)には最適のステートメントといえます。プログラム実行中に STOP キーを押した時も同様に止まった行番号を示して、プログラムはストップします。 ☞ CONT、RUN参照

サンプルプログラム

```
10 REM * STOP *
20 FOR I=1 TO 100
30 PRINT I
40 STOP
50 NEXT
RUN
 1
BREAK IN 40
OK.
CONT
 2
BREAK IN 40
OK.
■
```

（10～50：プログラム）
BREAK IN 40 ……行番号40のSTOPが実行される。
CONT ……CONT命令参照。
BREAK IN 40 ……行番号40のSTOPが実行される。

END

働き	プログラムの実行の終了を宣言します。
文法	END
省略形	E.
説明	ENDはプログラムの実行を終了させます。 このステートメントを実行するとプログラムを終了し、コマンド待ちの状態になります。 ENDはプログラムを終了させたいところに、どこにでも置くことができます。プログラムの最終行のENDは省略することができます。

サンプルプログラム

```
10 REM * END *
20 FOR I=1 TO 1000
30 PRINT I;
40 A$=INKEY$
50 IF A$="Z" THEN END
60 NEXT
RUN
 1 2 3 4 5 6 7 8
OK
■
```

Zキーを押したときにプログラムの実行は終了する。(8までプリントされたときにZキーを押した例です。

行番号40番でキー入力を受けつけ、行番号50番で入力キーの判断をします。Zキーが押されるとENDでプログラムの実行を終了しますが、それ以外の場合は、1000までの数値を連続して表示します。(ただし、STOPキーは、つねに受けつけます)

SWAP

働き	2つの変数の内容を交換します。
文法	SWAP 変数(1)，変数(2) 変数→内容を交換する変数（同じ型であることが必要です）
省略形	SW.
説明	SWAP文は、2つの変数の内容を交換します。たとえば、 SWAP A,B とすると、変数AとBの内容が交換されます。ただし、交換する変数の型は一致していなければなりません。 (SWAP A, B$は使えません)

サンプルプログラム

```
10 REM * SWAP *
20 DIM A(10)
30 FOR I=1 TO 10
40 READ A(I)
50 PRINT A(I);
60 NEXT
70 FOR I=1 TO 10
80 FOR J=1 TO 10
90 IF A(I)<A(J) THEN SWAP A(I),A(J)
100 NEXT
110 NEXT
120 PRINT
130 FOR I=1 TO 10
140 PRINT A(I);
150 NEXT
160 DATA 2,3,5,1,7,4,8,9,6,0
RUN
 2  3  5  1  7  4  8  9  6  0
 0  1  2  3  4  5  6  7  8  9
OK
■
```

配列変数の中にデータを読み込み、これを小さい順に並び変えるもので、ソーティングの基本となるものです。

REM

働き	プログラム文にコメントを入れます。
文法	REM〔コメント〕 コメント→任意の文字列（メッセージ）255文字まで GOSUB文のサブルーチンに使用する場合は 「'」（アポストロフィ）を使用ください 「REM」は使用できません
省略形	'（アポストロフィ）
説明	プログラムにコメントを挿入するためのステートメントで、プログラムの実行にはかかわりありません。 プログラム中どこでも書けます。 REMは、'（アポストロフィ）で代用できます。 REM文で書かれたプログラム行番号は、何も実行されません。(プログラム用メモリは使われます)

サンプルプログラム

```
10 ' * REM *
20 REM
30 REM NS-HUBASIC V1.0
40 '   SAMPLE PROGRAM
50 '
60 REM   「REM」=「'」
70 REM
RUN
OK
■
```

READ

働き　DATAで用意されたデータをREAD文の変数に入力します。

文法　**READ　変数［，変数，変数，……］**
　変数→DATA文のデータを入れる変数

省略形　**REA.**

説明　データの入力にはINPUT文（INPUT参照）ともう1つREAD文による方法があります。INPUT文は、プログラム実行時にデータを入力しますが、実行のたびに決まったデータを用いる場合など不便なので、このような場合にはREAD文を使うようにします。
READ文はDATA文（DATA参照）といつも対にしていっしょに使うステートメントで、READのあとに変数を書き、DATAのあとにはそれに対応する定数データを書きます。
READ文の変数とDATA文の定数データは1対1に対応しかつ双方とも同じ型でなければなりません。
DATA文はプログラム中のどこにあってもかまわず、1つのREAD文で2つ以上のDATA文からデータを読み込んだり、いくつかのREAD文の1つのDATA文を共用して読み込むこともできます。いずれの場合もDATA文は行番号の若い順に、データの並びの先頭から読み込まれます。
DATA文のデータの数がREAD文の変数の数より多い場合は、次に来るREAD文によって引き続いて読み込まれ、READ文がなければ残りのデータは無視されます。逆に、DATA文のデータの数が不足している場合は、「OD ERROR」のエラーメッセージが出ます。
RESTORE(RESTORE参照)を使うと、同じDATA文を何度でも読み直したり、読み込むDATA文の行番号を変えたりすることができます。
☞DATA、RESTORE参照

サンプルプログラム

(1)数値データの読み込み

```
10 REM * READ --- (1) *
20 FOR I=1 TO 10
30 READ X
40 PRINT X;
50 NEXT
60 DATA 3,4,1,6,2,7,8,3,4,9
RUN
 3  4  1  6  2  7  8  3  4  9
OK
■
```

(2)文字データの読み込み

```
10 REM * READ --- (2) *
20 READ A$,B$,C$
30 PRINT A$;" ";B$;"."
40 PRINT A$;" ";C$;"."
50 DATA GOOD,MORNING,EVENING
RUN
GOOD MORNING.
GOOD EVENING.
OK
■
```

(3)配列への読み込み

```
10 REM * READ --- (3) *
20 DIM A(5)
30 FOR I=0 TO 5
40 READ A(I)
50 PRINT A(I);
60 NEXT
70 DATA 9,1,8,3,4,8
RUN
 9  1  8  3  4  8
OK
■
```

DATA

働き　READで読み込むデータを用意します。

文法　**DATA　定数［，定数，定数，……］**
　定数→数値定数(−32768〜+32767)、文字定数（文字列）のデータ

省略形　**D.**

説明　DATA文はREAD文(READ参照)で読み込むデータを用意するステートメントです。DATA文によって用意できるデータは数値定数および文字定数で、1行につき255文字分入力可能です。
何かを実行するステートメントではないので、プログラム中のどこにでも、いくつでも置くことができますがプログラムの読みやすさという点から、READ文のすぐ下に書くか、プログラムの最後にまとめて書くのが最も一般的です。
文字定数（文字列）をデータとして用意する場合「"」（ダブルクォート）で囲まなくてもかまいません。ただし、「，」（カンマ）や「：」（コロン）は区切り記号として使うので、「，」や「：」をデータにする場合は、「" 」で囲む必要があります。
☞READ、RESTORE参照

サンプルプログラム

☞READ参照

```
DATA  ABC, DE, ",", F
       ↓    ↓   ↓   ↓
      ABC  DE   ,   F
```

RESTORE

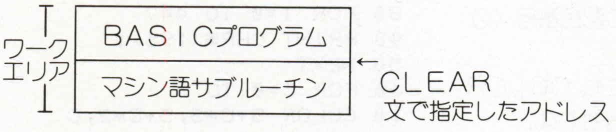
働き	READ文で読むDATA文を指定します。
文法	RESTORE［行番号］ 行番号→読み始めるDATA文の行番号
省略形	RES.
説明	BASICインタープリタは、DATAを指すポインタを持っており、READ文を実行すると、そのプログラムの先頭からDATA文を探して行き、最初に出てきたDATAにポインタをセットします。RESTOREを実行すると、DATA用のポインタが指定した行番号の行にセットされます。RESTOREのみのときは、ポインタがプログラムの先頭にセットされます。 RESTORE文を使えば、同じDATA文を何度でも利用したり、読み込むDATA文を任意に指定することができます。 ☞READ、DATA 参照

サンプルプログラム

```
10 REM * RESTORE *
20 RESTORE 1010
30 FOR I=0 TO 5
40 READ A
50 PRINT A;
60 NEXT
70 PRINT
80 RESTORE 1000
90 FOR I=0 TO 5
100 READ A
110 PRINT A;
120 NEXT
130 REM
1000 DATA 23,43,55,65,42,9
1010 DATA 12,56,34,68,53,2
RUN
  12  56  34  68  53  2
  23  43  55  65  42  9
OK
■
```

CALL

働き	マシン語サブルーチンを直接呼び出します。
文法	CALL アドレス アドレス→サブルーチンの実行開始番地
省略形	CA.
説明	アドレスを実行開始番地とするマシン語サブルーチンを呼び出します。 アドレスには16進定数（&H○○○○の形）、10進整数定数（−32768〜＋32767）などの整数定数、整数変数、式が指定されます。マシン語サブルーチンは、CLEAR文で指定したアドレス以降に置いてください。



POKE

働き	メモリ上に1バイトのデータを書き込みます。（⇔PEEK）
文法	POKE アドレス，データ〔，データ，データ，……〕 アドレス→データを書き込むアドレス（104ページのメモリマップをご覧ください） データ→0〜255までの整数
省略形	PO.
説明	メモリ上の指定したアドレスに直接1バイト（8ビット）のデータを書き込みます。データは、0〜255（&H0〜&HFF）の値でなければなりません。データを「，」（カンマ）で区切って続けて書くと、アドレスで指定したアドレスから連続したアドレスに書き込むことができます。 ただし、POKEは、現在のメモリの内容を書き換えてしまうため、不要意に使うとファミリーベーシックのシステム領域を破壊してしまいますので、十分に気をつけてください。 POKEで使用できるメモリの範囲は、&H7040〜77FFです。&H7000〜703Fはシステムで使用していますので、使用しないでください。

サンプルプログラム

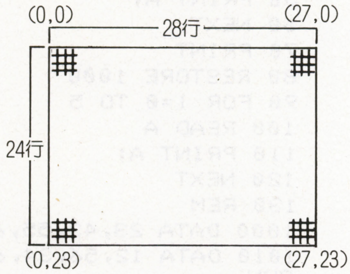
```
10 REM * POKE *
20 CLEAR &H7600
30 D=0
40 FOR A=&H7600 TO &H761F
50 POKE A,D
60 D=D+1
70 NEXT
80 FOR A=&H7600 TO &H761F
90 RD=PEEK(A)
100 PRINT "  ";HEX$(RD);
110 NEXT
```

行番号40〜70番のループで、メモリの7600〜761Fの各アドレスに0〜31の数値を書き込んでいます。80〜110番のループでは、各アドレスに実際にデータが入っているかどうか調べるため、PEEK関数（85ページ）を使ってデータを読み込み16進数で表示しています。

画面制御ステートメント

LOCATE

働　き	カーソルを指定位置へ移動します。
文　法	LOCATE X, Y X……水平方向の表示カラム　0~27 Y……垂直方向の表示行　0~23
省略形	LOC.
説　明	LOCATEはバックグラウンド面のカーソル位置を指定します。



サンプルプログラム

```
10 REM * LOCATE *
20 CLS
30 FOR I=0 TO 20
40 LOCATE I,I:PRINT"*";
50 NEXT
60 LOCATE 0,10
```

行番号40番でカーソル位置を指定、＊をプリントします。行番号60番でカーソル位置を指示位置に移動し、プログラムの実行を終了します。（終了位置より命令を与えることはできません。カーソルを＊表示のない行まで移動してください）

COLOR

働　き	バックグラウンド面へ表示する文字の配色番号を画面上のエリア毎に指定します。
文　法	COLOR X, Y, n X…水平方向の表示カラム　0~27 Y…垂直方向の表示行　0~23 n…配色番号　0~3
省略形	COL.
説　明	X、Yで指定された位置を含む画面上のエリア毎に、背景、または文字の表示色をCGSETで指定されたカラーパレットコード内の配色番号の中から選びます。 113ページのカラーチャートをご覧ください。 BG GRAPHICは、カラーパレット1の配色番号（0~3）で配色表示されています。 CGSETでカラーパレットコードを変更しても、COLOR文で指定したエリア内の配色番号はそのまま有効です。

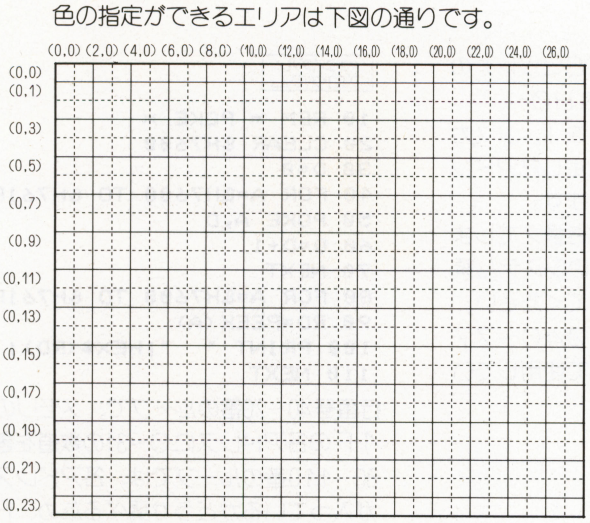
サンプルプログラム

☞CGSET参照

```
10 REM * COLOR *
20 CLS
30 FOR I=0 TO 447
40 PRINT CHR$(195);
50 NEXT
60 FOR C=0 TO 3
70 COLOR 5+C*3,5+C*2,C
80 NEXT
90 LOCATE 0,20
```

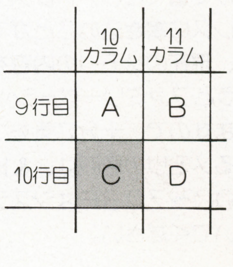
行番号70番で指定された位置を含むエリアを、指定した配色番号によって配色表示します。行番号90番でカーソルを指示位置に移動し、プログラムの実行を終了します。

色の指定ができるエリアは下図の通りです。



COLORは、□で示した範囲のエリア毎に配色番号を指定できます。

例、COLOR 10，10，3 とすれば



X、Yで指定した（10，10）を含む画面上のエリア内（A，B，C，D）にある4つのキャラクタ（文字、記号、グラフィックキャラクタ）は、3の配色番号の色で表示されます。

CGEN

働　き	バックグラウンド面とスプライト面にキャラクタの割り当てを決めます。

文　法 CGEN n
n…割り当ての組み合わせ　0～3

省略形 CGE.

説　明 キャラクタテーブルA（裏表紙）やB（113ページ）で用意されている、マリオ等のアニメキャラクタ（スプライト用）と数字・英文字・記号・カナ・背景パターン（BG GRAPHIC用）のキャラクタを、バックグラウンド面、スプライト面のどちらの面で使用するかを決めます。バックグラウンド面にマリオ等のアニメキャラクタを表示したり、文字記号をスプライトとしてスプライト面に表示することができます。

キャラクタテーブル A	キャラクタテーブル B
マリオなどのアニメキャラクタ	数字・英文字・記号・カナを含む背景パターン

106～109ページのキャラクタコード表A、Bをご覧ください。

n	バックグラウンド面	スプライト面	意　味
0	A	A	BG面とスプライト面の両方でキャラクタテーブル Aのキャラクタを使用します。
1	A	B	BG面にキャラクタテーブル Aのキャラクタ、スプライト面にキャラクタテーブル Bを使用します。
2	B	A	BG面にキャラクタテーブル Bのキャラクタ、スプライト面にキャラクタテーブル Aのキャラクタを使用します。
3	B	B	BG面とスプライト面の両方でキャラクタテーブル Bのキャラクタを使用します。

デフォルト値…CGEN2
CTR＋Dキーでこ CGENは2の値に再設定されます。

サンプルプログラム

```
10 REM * CGEN *
20 CLS:SPRITE ON:CGSET 0,1
30 FOR I=32 TO 255
40 PRINT CHR$(I);
50 NEXT
60 DEF SPRITE 0,(0,1,0,0,0)=CHR$(64)
   +CHR$(65)+CHR$(66)+CHR$(67)
70 SPRITE 0,100,150
80 PAUSE 100:BEEP
90 CGEN 0
100 PAUSE 100:BEEP
110 CGEN 1
120 PAUSE 100:BEEP
130 CGEN 3
140 PAUSE 100:BEEP
150 CGEN 2
```

（30～40）画面上に文字記号の一覧を表示する。
（60）アキレス(左1)を定義する。

行番号 10―80 … CGEN2で表示されている。
行番号 90―100 … CGEN0で表示
行番号110―120 … CGEN1で表示
行番号130―140 … CGEN3で表示
行番号150 … CGEN2で表示に戻す。

CLS

働　き 画面をクリアします。

文　法 CLS

省略形 CL

説　明 バックグラウンド面をクリアします。
（バックグラウンド面に、コピー（複製）されたBG GRAPHICも同時に消えます。プログラム上でBG GRAPHICをバックグラウンド面にコピー（複製）する場合は、CLS命令と入れ替えてVIEW命令を使用してください。

サンプルプログラム

☞LOCATE文参照

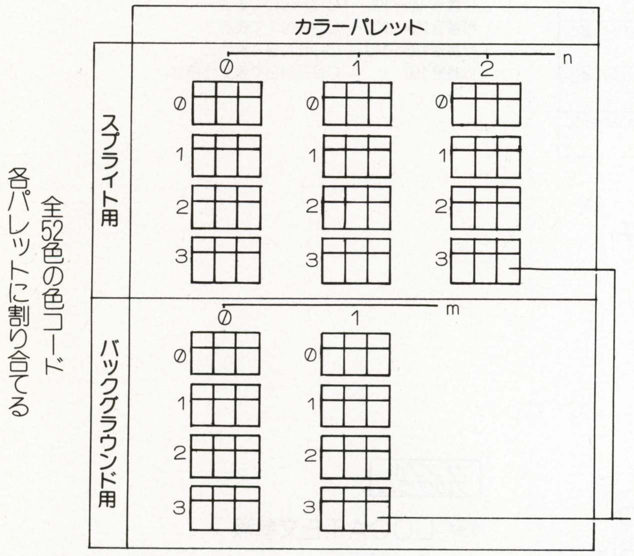
CLS RETURN……画面はクリアされ画面左上にOK表示がでます。

```
OK
■
```

10 CLS……画面をクリアした後、次の行番号のプログラムを実行します。

CGSET

働　き	BGやスプライトで使用するパレットの割り当てを決めます。 (表示される色の組み合わせを指定する)
文　法	CGSET ［m］［, n］ m…バックグラウンド面用のパレットコード　0～1 n…スプライト（アニメキャラクタ）用のパレットコード　0～2
省略形	CG.
説　明	あらかじめ用意されている色の組み合わせのグループの中から選んで、背景の色やアニメキャラクタの表示色を決めます。表示可能な全52色の中からバックグラウンド用とスプライト用にカラーチャート（113ページ）で示されているように、それぞれのカラーパレット上へ色（色コード）が設定されます。バックグラウンド用で2種類（パレットコード0～1）、スプライト用で3種類（パレットコード0～2）のカラーパレットが準備されています。それぞれのカラーパレットには配色番号（0～3）の各グループに各々3種類ずつ、計12色の色コードが保持できます。 一度指定すると、再度指定し直すか、PALET命令で配色の色コードを変えるまで有効となります。 〔バックドロップ面の色は黒（トウメイ）になります〕



VIEW命令でBG　GRAPHIC面に描いた絵をそのままの色で、バックグラウンド面にコピーするには、バックグラウンド面のパレットコードは1（CGSET　1，1)を使用してください。

デフォルト値…m＝1、n＝1
113ページのカラーチャート（CGSET）をご覧ください。
CTR＋Dを操作する、またはGAME　BASICモード画面から、1－－BASICと選び、BASIC面になった時、カラーパレットはバックグラウンド用、スプライト用ともにバックグラウンド用のパレットコード1の配色となっています。

サンプルプログラム

```
CGSET 0,0
OK
■
10 CGSET 0,1
```

DEF　SPRITEやDEF　MOVE文で指定する配色番号は、表示されるスプライトの色をCGSETで決められたパレット中の配色番号（0～3）の色の組み合わせを指定することです。
例えば、
CGSET　1，1　スプライト用パレットコードの指定
DEF　SPRITE　0，（0，1，0，0，0）＝…
DEF　MOVE（N）＝SPRITE（0，3，1，10，0，3）

上のCGSETで指定したスプライト用パレット1の中の配色番号を指定（それぞれ0と3）
スプライトは、この色で配色表示される。
バックグラウンド面の配色番号指定はCOLR文で行ないます。
70ページをご覧ください。

70ページをご覧ください。

PALET

働き 配色番号内の色コードを任意の色コードに設定し直します。

文法 PALET {B/S} n, C1, C2, C3, C4

B…BG（背景）
S…スプライト（アニメキャラクタ）
n…配色番号　0～3
C1…バックドロップ面の色を指定します
　　nの値が0のとき有効となります

C1, C2, C3, C4 …52色の色コードを指定します。
C2、C3、C4は配色番号内の
左端、中央、右端の各色に対応します。
例
配色番号0　C2 C3 C4
色コードは下表の値（0～60）を使用します。

省略形 PAL.B
PAL.S

説明 背景の色やアニメキャラクタの表示色を、52色の色コードの中から選んで着色します。バックドロップの色は、バックグラウンドで使用しても、スプライトで使用しても画面に表示着色されます。（PALETB、PALETSに関係しない）プログラム上では実行行番号の大きなプログラムで指定した色が最終的に表示されます。
CGSET文で決められたパレットに対して実行されます。さらに、そのパレット内のカラーグループに対応するnの値を決め、4色（C1, C2, C3, C4）のコードを設定します。色コードはカラーゼネレータで発色される52色の中から選択できます。

サンプルプログラム

```
10 REM * PALET *
20 SPRITE ON
30 DEF SPRITE 0,(3,1,0,0,0)=CHR$(88)
   +CHR$(89)+CHR$(90)+CHR$(91)
40 SPRITE 0,100,150
RUN
OK      …スプライト用パレットコード1で配色番号3の色で表示される。
CGSET 1,0
OK      …スプライト用パレットコード0で 配色番号3の色で表示される。
PALETS 3,&H0F,&H30,&H26,&H12 …&H30，&H26，&H12の色で表示される。
OK
■
```

《52色の色コード》

色相	16進	10進	16進	10進	16進	10進	16進	10進	
	00	0	10	16	20	32	30	48	灰色～白色
青系	1	1	11	17	21	33	31	49	
	2	2	12	18	22	34	32	50	
	3	3	13	19	23	35	33	51	
	4	4	14	20	24	36	34	52	
赤系	5	5	15	21	25	37	35	53	
	6	6	16	22	26	38	36	54	有色
	7	7	17	23	27	39	37	55	
	8	8	18	24	28	40	38	56	
緑系	9	9	19	25	29	41	39	57	
	A	10	1A	26	2A	42	3A	58	
	B	11	1B	27	2B	43	3B	59	
	C	12	1C	28	2C	44	3C	60	
	0D	13	1D	29	2D	45			クロ
	0E	14	1E	30	2E	46			
	0F	15	1F	31	2F	47			

暗←→明

CGSET 1,0 ─スプライトのパレットコード0を指定。
DEF SPRITE 0,(0,1,0,0,0)="@ABC"
└パレットコード0の中の配色番号0を指定

スプライトは、この時色コード54、22、2の色を配色表示されます。

PALETS 0,1,48,25,18

- バックドロップの色を1にする。
- 54の色を48の色にする
- 22の色を25の色にする。
- 2の色を18の色にする。

ムーブコマンド

アニメキャラクタをスプライト面上に表示、動作させる命令群です。DEF SPRITEや SPRITE n，x，yと同格ですが、ゲーム表現をより簡単に行なうことができる特長ある命令です。

DEF MOVE

働き　アニメキャラクタに対して固有の動きを定義します。
(扱えるアニメキャラクタの種類は全部で16種類です)

文法　DEF MOVE(n)=SPRITE(A,B,C,D,E,F)

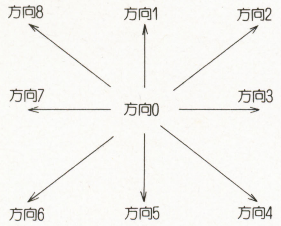
n…アニメキャラクタの動作番号　0~7
A…アニメキャラクタの種類　0~15
B…動作の方向指定　0~8
C…動かす速さ　1~255(0の場合、256フレーム毎に動作します) 1…最高速　255…最低速
D…全移動量　1~255 (0とすると表示されません)
E…表示優先度　0~1 (0…バックグラウンドより前のスプライト面に表示)
(1…バックグラウンドより後ろのスプライト面に表示)
F…配色番号　0~3

省略形　DE.M.

説明　16種類の中からアニメキャラクタを8種類まで選び、その動きを定めます。

- アニメキャラクタの動作番号は、=以後で定義したアニメキャラクタの固有の動きに対してつけられる番号です。
- アニメキャラクタの種類は次のアニメキャラクタが割り当てられています。(裏表紙のキャラクタテーブルAをご覧ください)

0：マリオ	9：スターシップ
1：レディ	10：爆発
2：ファイターフライ	11：ニダニタ
3：アキレス	12：レーザー
4：ペンペン	13：シェルクリーパー
5：ファイアー ボール	14：サイドステッパー
6：車	15：ニットピッカー
7：スピナー	
8：スターキラー	



動作の方向は、時計回りで8方向に対応しています。
0を指定すると停止しているキャラクタになります。

- 動く速さは、指定の動かす速さ(C)の2倍のフレーム毎(約1/30秒)に2ドットずつ(ななめの方向の場合は、X、Yそれぞれ2ドット)動き、速さ(ドット/秒)=2ドット/(C×1/30秒)=60/C(1ドット/秒)で表わされます。
例えば、1を指定すると、速さは1秒で60ドット動き、255の場合、速さは255秒で60ドット動きます。
- 全移動量は、アニメキャラクタを動かす距離を指定します。指定した2倍のドット数だけ、アニメキャラクタは移動します。全移動ドット数=2×D(ドット)(ななめ方向の場合は、X、Yとも2×Dドット)となります。
- 表示優先度は、アニメキャラクタをバックグラウンド面の前のスプライト面に表示するか(0)、バックグラウンド面の後のスプライト面に表示するか(1)を指定します。
- 配色番号は表示する配色をCGSETで指定したパレットコードの中の配色番号(0~3)で指定します。

MOVE

働き　アニメキャラクタの動きを開始させます。

文法　MOVE n_0[, n_1 , n_2 , n_3 , n_4 , n_5 , n_6 , n_7]

n_0〜n_7…DEF　MOVEで定義した動作番号　0〜7

省略形　M.

説明　あらかじめDEF　MOVE命令で定義されたアニメキャラクタの動きを開始させます。（SPRITE　ONでスプライトを表示するようにしておいてください）
アニメキャラクタの動きを同時に8個まで開始させることができます。
MOVE文が実行されると、DEF　MOVE命令で指定された動きが完了するまで、BASICのコントロールとは非同期にアニメキャラクタは動きます。つまり、プログラムが終了している場合でも、アニメキャラクタは、DEF　MOVE文で定義された移動量、方向指定、速度に応じ、動きが完了するまで動きます。
スプライト面上への表示は、同じ水平方向上には4個まで表示されます。

CUT

働き　MOVE文で動きを開始させたアニメキャラクタの動きを停止させます。

文法　CUT n_0 [, n_1 , n_2 , n_3 , n_4 , n_5 , n_6 , n_7]

n_0〜n_7…DEF　MOVEで定義した動作番号　0〜7

省略形　CU.

説明　MOVE文で動きを開始させたアニメキャラクタの中から任意の動作番号の動きを停止します。同時に8個まで指定できます。
CUTした後、再度同じ動作番号のアニメキャラクタをMOVEで動きを再開させた場合は、あらかじめDEF　MOVEで定義されていた全移動量（D）の残りの移動量を完了するまで、止まっていた位置から動き始めます。

サンプルプログラム

```
CUT 0,1,2
OK
■
 10 CUT 0,1,2,3,4,5,6,7
```

ERA

働き　MOVE文で動きを開始させたアニメキャラクタの動きを停止させ、スプライト面上から消します。

文法　ERA　n_0 [, n_1 , n_2 , n_3 , n_4 , n_5 , n_6 , n_7]

n_0〜n_7…DEF　MOVEで定義した動作番号　0〜7

省略形　ER.

説明　MOVE文で動きを開始させたアニメキャラクタやCUT文で停止している(または動きが完了した)アニメキャラクタの中から任意のものの表示を消します。同時に8個まで指定できます。
ERAした後、再度同じ動作番号のアニメキャラクタをMOVEで動きを再開させた場合は、あらかじめDEF　MOVEで定義されていた全移動量(D)の残りの移動量を完了するまで、消えた位置から現われて動き始めます。

サンプルプログラム

```
ERA 0,1,2
OK
■
 10 ERA 0,1,2,3,4,5,6,7
```

POSITION

働き　MOVE文で動きを開始させる前に、動き始めの初期座標を与えます。(スプライト面の座標)

文法　POSITION n,X,Y

n…DEF　MOVEで定義した動作番号　0~7
X…水平方向の座標　0~255
Y…垂直方向の座標　0~255
(ただしスプライト面上有効な範囲はX:0~240, Y:5~220)
(※テレビ受像機によって画面の有効表示範囲は異なります。)

省略形　POS.

説明　MOVE文でアニメキャラクタの動きを開始させる前に、この命令で動作を開始する初期座標を与えます。X、Yの値は、アニメキャラクタの左上端が位置するスプライト面上の座標です。
指定しない場合、デフォルト値…X=120、Y=120の値で動きを開始します。
プログラム中で同じ動作番号のアニメキャラクタをMOVEで、くり返し動きを開始させた場合は、DEF　MOVEで定義され動きを完了した位置を、次の動きの初期座標として動きを開始します。
RUNまたは再度POSITION命令で動きの初期座標を指定し直すか、DEF　MOVEで動きを再定義するまで同じ動作番号のアニメキャラクタは、前に動きを完了した座標位置を保持しています。

サンプルプログラム

```
10 REM * POSITION *
20 CLS:SPRITE ON
30 DEF MOVE(0)=SPRITE(11,3,2,10,1,2)
40 X=RND(256):Y=RND(240)
50 PRINT"X,Y=";X;",";Y
60 POSITION 0,X,Y
70 MOVE 0
80 PAUSE 80
90 GOTO 20
```

行番号30…ニタニタを右方向へ20ドット動かす様定義します。
行番号40…X, Yを乱数で発生します。
行番号50…X, Yの値を表示します。
行番号60…ニタニタの動きを開始させる位置を指定します。
行番号70…ニタニタの動きを開始させます。

XPOS

働き　DEF　MOVEで定義された動作番号の位置の、水平方向の座標値を求めます。

文法　XPOS(n)

n…DEF　MOVEで定義されたアニメキャラクタの動作番号　0~7

省略形　XP.

説明　XPOS関数が実行されたとき、nで指定された動作番号のアニメキャラクタ水平方向の座標値を求めます。
動作を順次移行する場合には、POSITION文と組み合わせて使用します。

サンプルプログラム

```
10 REM * XPOS  YPOS *
20 CLS:SPRITE ON
30 DEF MOVE(0)=SPRITE(0,2,2,10,0,0)
40 MOVE 0
50 LOCATE 8,20:PRINT"      "
60 LOCATE 8,21:PRINT"      "
70 LOCATE 0,20:PRINT"XPOS(0)=";XPOS(0)
80 LOCATE 0,21:PRINT"YPOS(0)=";YPOS(0)
90 PAUSE 50
100 GOTO 40
```

行番号30…マリオをX, Yそれぞれに20ドットずつななめ右上へ動かす様、定義します。
行番号70－80…マリオのXPOS, YPOSの値を表示します。

YPOS

働き　DEF　MOVEで定義された動作番号のアニメキャラクタ位置の、垂直方向の座標値を求めます。

文法　YPOS(n)

n…DEF　MOVEで定義されたアニメキャラクタの動作番号　0~7

省略形　YP.

説明　YPOS関数が実行されたとき、nで指定された動作番号のアニメキャラクタの垂直方向の座標値を求めます。
動作を順次移行する場合には、POSITION文と組み合わせて使用します。

サンプルプログラム

☞XPOS参照

MOVE(n)

働き　MOVE文で動きを開始させた動作番号のアニメキャラクタが、DEF　MOVEで定義した動きを完了したかどうかをこの関数に値で与えます。

文法　MOVE(n)

n…DEF　MOVEで定義したアニメキャラクタの動作番号　0～7

省略形　M.(n)

説明　MOVE文で動きを開始させたアニメキャラクタの動きに対応して、この関数に与えられる値が異なります。
MOVE(n)関数は、指定した動作番号のアニメキャラクタが、DEF　MOVEで定義された動きが完了していれば0、動きを行なっている場合は−1の値で与えられます。

サンプルプログラム

```
10 REM * MOVE(N) *
20 SPRITE ON :CGSET 1,0
30 DEF MOVE(0)=SPRITE(0,3,1,150,0,0)
40 MOVE 0
50 IF MOVE(0)=-1 THEN PRINT"MOVE(0)=
   ";MOVE(0):GOTO 50
60 PRINT"MOVE(0)=";MOVE(0)
70 END
```

行番号30…マリオを右方向へ300ドット動かすよう定義します。
行番号50…MOVE(0)の値を判別し、マリオが動きを完了したかどうかを判断しています。
行番号60…マリオが動きを完了したときMOVE(0)=0と表示します。

特殊ステートメント

KEY

働　き　ファンクションキーに対して文字列を定義します。

文　法　**KEY　ファンクションキー番号，文字列**

ファンクションキー番号……1～8（ファンクションキー左端から1、2、3、4、5、6、7、8と対応しています）

文字列………………………最大15文字まで定義することができ、15文字を越えるとはみ出した右側の文字が無視されます。

省略形　K.

説　明　ファンクション番号で指定したキーに文字列を対応させます。このステートメントによって文字列を定義すると、それ以降、そのキーを押すと定義した文字列をキーボードから打ち込むのと同じになります。ファンクションキーは、GAME BASICモード画面から、BASIC面を選んだ時には次のように定義されています。

F1. LOAD(M)
F2. PRINT
F3. GOTO
F4. CHR$(
F5. SPRITE
F6. CONT(M)
F7. LIST(M)
F8. RUN(M)

（M）はRETURNのことです。（M）を定義するためには文字列の後ろに+CHR$（13）を入力してください。

☞KEY LIST参照

サンプルプログラム

```
KEYLIST                      …ファンクションキーの内容を表示する。
F1 LOAD(M)                   …（M）はRETURNのこと
F2 PRINT
F3 GOTO
F4 CHR$(
F5 SPRITE
F6 CONT(M)
F7 LIST(M)
F8 RUN(M)
OK
KEY1,"SAVE"+CHR$(13)         …F1にSAVE RETURNを定義する。CHR$（13）はRETURN
OK
KEY5,"DEF MOVE("             …F5にDEF MOVE（を定義する
OK
KEYLIST
F1 SAVE(M)
F2 PRINT
F3 GOTO                      …F1とF5に定義されている
F4 CHR$(
F5 DEF MOVE(
F6 CONT(M)
F7 LIST(M)
F8 RUN(M)
OK
■
```

（F1～F8：デフォルト値）

KEYLIST

働　き　ファンクションキーの定義状態を画面に表示します。

文　法　KEY　LIST

省略形　K.L.

説　明　このステートメントを実行すると、画面にファンクションキー番号と文字列の一覧表を表示します。

☞KEY参照

PAUSE

働　き　プログラムの実行を一時休止します。

文　法　PAUSE [n]

n…0～32767

省略形　PA.

説　明　このステートメントを実行すると、設定単位時間分プログラムの実行を休止したのち次に進みます。

nを省略すると、何かのキー入力があるまでプログラムの実行を休止しています。キー入力があると次のプログラムへ進みます。

サンプルプログラム

```
10 REM * PAUSE *
20 FOR I=0 TO 10
30 BEEP:PAUSE 10
40 NEXT
50 PAUSE                     …何かキー入力があるまでここでプログラムの進行を休止している。
60 FOR I=0 TO 10
70 PLAY"C":PAUSE 20
80 NEXT
```

SYSTEM

働き	BASIC実行モードからGAME　BASICモード画面へ戻します。
文法	SYSTEM
省略形	S.
説明	BASICの面で、SYSTEMコマンドを入力すると、GAME BASICモード画面にかわります。プログラム上では使用しないでください。プログラムは保護されたままです。1－－BASICを選択してBASICの面に再度入った時、続けてBASICができます。

サンプルプログラム

```
SYSTEM        …ダイレクト命令
```

VIEW

働き	BG　GRAPHIC面をバックグラウンド面へコピー（複製）します。
文法	VIEW
省略形	V.
説明	BASICを実行中、VIEWコマンドを入力すると、BG GRAPHIC面に描いた絵がバックグラウンド面へコピー（複製）されます。バックグラウンド面にコピーされたBG GRAPHICはBASICで書きかえができますが、BG　GRAPHIC面にはその絵が残っています。 BG　GRAPHIC面に描いた絵をそのままの色で、バックグラウンド面にコピー表示するためには、バックグラウンド面のパレットコードは1（CGSET　1，1）を使用してください。

サンプルプログラム

```
VIEW          …ダイレクト命令
               （カーソルはホーム位置へ戻る）

10 VIEW       …プログラム中
```

サウンド制御ステートメント

BEEP

働　き	「ピー」という音を出します。
文　法	BEEP
省略形	B.

サンプルプログラム

☞PAUSE、IF～THENをご覧ください。

PLAY

働　き	音楽を演奏します。
文　法	PLAY　ストリングデータ ストリングデータ…以下で説明する文字列変数をストリングデータといいます
省略形	PL.

説　明　ストリングデータで指定された音に従って演奏します。ストリングデータは、音程、オクターブ、音の長さ、音量、和音、エンベロープ、デューティを指定するもので、以下に説明する文字を並べたものです。

チャンネルセパレータ（コロン）を使って3重音まで演奏できます。

①テンポ…T1～T8でテンポを指定します。
　T1(はやい)←――→T8(おそい)

②デューティ効果(音色)…Y0～Y3で音色を指定します。

Y0：12.5%	●デューティ効果はエンベロープ(M0またはM1)指定を実行した後有効となります。
Y1：25.0%	
Y2：50.0%	●M0またはM1を指定後、音色(Y0～Y3)を指定してください。
Y3：75.0%	

③エンベロープ…M0、M1でエンベロープをかけるか否かを指定します。エンベロープがかかると、余いんのある音、切れた音、はねた音になります。

●M0(エンベロープをかけない)のとき、VnでV0～V15で音量を指定します。

0(小)←――音量――→(大)15

M0V15 ―― 音量15となる

●M1(エンベロープをかける)のとき、VnはV0～V15でエンベロープの長さを指定します。

0(短い)←――エンベロープ――→(長い)15

M1V3 ―― エンベロープ3となる。音量はV15の音量となります。

④オクターブ…O0～O5でオクターブを指定します。
　O0(低音)←――→O5(高音)

⑤音程…音程は下記のようにC～Bの文字を使って表されます。

音　程	指定方法
ド	C
ド♯(レ♭)	♯C
レ	D
レ♯(ミ♭)	♯D
ミ	E
ファ	F
ファ♯(ソ♭)	♯F
ソ	G
ソ♯(ラ♭)	♯G
ラ	A
ラ♯(シ♭)	♯A
シ	B

サンプルプログラム

☞RND参照

```
PLAY"CRDRE"
OK
PLAY"T4Y2M0V15O3C5R5D5R5E5"
OK
PLAY"T2Y0M1V9O1C3R5D6R1E4"
OK
PLAY"C:E:G"  …重音(3音同時出力)はコロンで区切る
OK        (C:E 中高音　G 低音)
PLAY"O5C5:O4E5:O1G5"
OK
■
```

3音同時出力の場合、各チャンネルごとに各パラメータを指定する。

一度指定するとT，Y，M，V，Oはずっと値を維持します。

T(テンポ)、Y(デューティ効果)、M(エンベロープ)、V(エンベロープの長さまたは音量)、O(オクターブ)

```
PLAY"T4Y2M0V15O3C5R5D5R5E5"
```

T4：テンポ　Y2：デューティ　M0V15：エンベロープ　O3：オクターブ　C：音程　5：長さ　R：休符　5：長さ

⑥休符…R0～R9で、休みの長さを指定します。
⑦長さ…音程及び休符、記号の後に0～9の整数をつけて、
音や休符の長さを指定します。

音　の　長　さ		対応する整数
$\frac{1}{8}$（32分音符𝅘𝅥𝅰　）	𝅀	0
$\frac{1}{4}$（16分音符𝅘𝅥𝅯　）	𝄿	1
$\frac{3}{8}$（付点16分音符𝅘𝅥𝅯.）	𝄿·	2
$\frac{1}{2}$（8分音符♪　）	𝄾	3
$\frac{3}{4}$（付点8分音符♪.）	𝄾·	4
1（4分音符♩　）	𝄽	5
$1\frac{1}{2}$（付点4分音符♩.）	𝄽·	6
2（2分音符𝅗𝅥　）	𝄼	7
3（付点2分音符𝅗𝅥.）	𝄼·	8
4（全音符𝅝　）	𝄻	9

注）音の長さは4分音符（整数5）を1としたときの相対的な値です。（C5R1のように書きます）
整数の指定がない場合は前に書かれた音と同じ長さになります。

⑧重音……2重音や3重音を演奏するには、チャンネルセパレータ：（コロン）をはさみます。
" チャンネルA：チャンネルB：チャンネルC"
各チャンネルごとに、音程、音の長さ、テンポ、エンベロープ、デューティを指定する必要があります。
ただしチャンネルCはエンベロープ、デューティは変わりません。
デフォルト値…T4、M0、V15、O3、音の長さ及び休符の長さ5。
1度指定すると、再度指定するまで各パラメータの値は継続します。

関数

数値関数

ABS

働き　数式の絶対値を与えます。

文法　ABS(x)

x…絶対値を求める数式（整数－32768～＋32767）

省略形　AB.

説明　数式xの絶対値｜x｜が、この関数の値になります。

サンプルプログラム

```
PRINT ABS(41)
 41
OK
PRINT ABS(-41)
 41
OK
PRINT ABS(10-34)
 24
OK
■
```

SGN

働き　数式の符号を与えます。

文法　SGN(x)

x…数式（整数－32768～＋32767）

省略形　SG.

説明　数式xによって、次の値がこの関数の値になります。

x＞0　のとき　1
x＝0　のとき　0
x＜0　のとき　－1

サンプルプログラム

```
PRINT SGN(41)
 1                ……正である。
OK
PRINT SGN(0)
 0                ……0である。
OK
PRINT SGN(-41)
-1                ……負である。
OK
■
```

RND

働き　引数未満の乱数（整数）を発生します。

文法　RND　(x)

x…数式（整数）1～32767

省略形　RN.

説明　発生した整数の乱数が、この関数の値になります。この乱数は、「引数(x)－1」の乱数になっています。
RND(1)は常に0です。

サンプルプログラム

RNDによる乱数で音楽を演奏してみましょう。

```
10 REM * RND *
20 PLAY"M1V6T201"
30 A$="CDEFGABR"
40 N=RND(8)
50 B$=MID$(A$,N+1,1)
60 PRINT B$;
70 PLAY B$
80 GOTO 40
```

……A$の文字列から1文字取りだしてB$とする。取りだす文字は行番号40で発生した8未満（0～7）の乱数値を使用しています。
取り出した文字を表示しながら取り出した文字の音を出します。

文字関数

ASC

働き	文字コードを数値に変換します。（⇔CHR$） ASC（文字列）
文法	文字列…数値に変換する文字が左端にある文字列 （左端の文字のみが変換されます）
省略形	AS.
説明	文字列の最初の1文字のキャラクタコードが、この関数の値になります。キャラクタコードの値は0~255の整数値です。文字列は式や変数でかまいません。また、文字列がヌルストリングのときに、この関数の値は0になります。

サンプルプログラム

```
10 REM * ASC *
20 INPUT"ﾓｼﾞ ﾊ";A$         ……先頭1文字のみの変換。
30 A=ASC(A$)
40 PRINT A$;" ﾉ ｷｬﾗｸﾀｺｰﾄﾞ ﾊ";A
RUN
ﾓｼﾞ ﾊ?H                    ……Hを入力すると
H ﾉ ｷｬﾗｸﾀｺｰﾄﾞ ﾊ 72
OK
```

ASCコード対照表は102~105ページのキャラクタコード表をご覧ください。

CHR$

働き	数値をキャラクタコードとみなして、対応する文字に変換します。（⇔ASC）
文法	CHR$（x） x ……文字に変換する数式 各0~255
省略形	CH.
説明	数値をキャラクタコードとする文字が与えられます。数値1つにつき1文字が得られます。プリント表示される文字、記号はXの値が32から255です。

サンプルプログラム

```
10 REM * CHR$ *
20 INPUT"ｷｬﾗｸﾀｺｰﾄﾞﾊ(0-255)";A
30 A$=CHR$(A)
40 PRINT A;" ﾆ ﾀｲｽﾙ ﾓｼﾞﾊ ";A$
RUN
ｷｬﾗｸﾀｺｰﾄﾞﾊ(0-255)?65          ……65を入れると
 65 ﾆ ﾀｲｽﾙ ﾓｼﾞﾊ A
OK
■
```

VAL

働き	数字の文字列を数値に変換します。
文法	VAL（文字列） 文字列…数字の文字列　−32768~+32767 &H0~&H7FFF
省略形	VA.
説明	文字列の中の文字としての数字を数値に変換します。もし文字列の最初の文字が、+、−、&または数字でないときには、この関数の値は0になります。また、文字列の中に数字以外の文字（16進数の場合はA~Fを除く）が現れたら、それ以降の文字は無視します。

サンプルプログラム

```
10 REM * VAL *
20 INPUT"16ｼﾝｽｳｦ ﾆｭｳﾘｮｸｼﾃｸﾀﾞｻｲ";A$
30 V=VAL("&H"+A$)
40 PRINT A$,"&H";A$;"=";V
RUN
16ｼﾝｽｳｦ ﾆｭｳﾘｮｸｼﾃｸﾀﾞｻｲ?AD     ……ADを入力すると
AD       &HAD= 173
OK
■
```

STR$

働き	数式の値を文字の数字に変換します。
文法	STR$（x） x…数式
省略形	STR.
説明	数式および数値を表す文字列が、この関数の値になります。正の数の場合は先頭にスペースが1文字はいります。

サンプルプログラム

```
10 REM * STR$ *
20 INPUT"ｽｳﾁ A ﾊ";A
30 A$=STR$(A)           ……Aの数値を文字の数値に変換
40 INPUT"ｽｳﾁ B ﾊ";B
50 B$=STR$(B)           ……Bの数値を文字の数値に変換
60 PRINT A;B;A+B        ……数値としての表示
70 PRINT A$;B$;A$+B$    ……数字の文字としての表示
```

HEX$

働　き	数式を16進数の文字列に変換します。
文　法	HEX$(x) x…数式　−32768〜+32767
省略形	H.
説　明	数式を16進数の文字列に変換したものが、この関数の値になります。

サンプルプログラム

10進数 0〜20を16進数に直す換算表を表示します。

```
10 REM * HEX$ *
20 FOR I=0 TO 20
30 PRINT"&H";HEX$(I);"=";I
40 NEXT
```

LEFT$

働　き	文字列の左側から、指定した数だけ文字を取り出します。 (⇔RIGHT$)
文　法	LEFT$(文字列, n) 文字列……最大文字数31文字 n…文字列から取り出す文字数　0〜255
省略形	LEF.
説　明	文字列の左側から取り出されたn個の文字列が、この関数の値になります。nの値が文字列の文字数より大きいときには、すべての文字が、n=0のときには、ヌルストリングがそれぞれ関数の値になります。

サンプルプログラム

```
10 REM * LEFT$ *
20 A$="ｺﾝﾆﾁﾊ"
30 FOR I=1 TO 5
40 PRINT LEFT$(A$,I)
50 NEXT
RUN
ｺ
ｺﾝ
ｺﾝﾆ
ｺﾝﾆﾁ
ｺﾝﾆﾁﾊ
OK
■
```

RIGHT$

働　き	文字列の右側から、指定した数だけ文字を取り出します。 (⇔LEFT$)
文　法	RIGHT$(文字列, n) 文字列……最大文字数31文字 n…文字数　0〜255
省略形	RI.
説　明	文字列の右側から取り出されたn個の文字列が、この関数の値になります。nの値が文字列の文字数より大きいときには、すべての文字が、n=0のときには、ヌルストリングがそれぞれ関数の値になります。

サンプルプログラム

```
10 REM *RIGHT$ *
20 A$="ｺﾝﾆﾁﾊ"
30 FOR I=1 TO 5
40 PRINT RIGHT$(A$,I)
50 NEXT
RUN
ﾊ
ﾁﾊ
ﾆﾁﾊ
ﾝﾆﾁﾊ
ｺﾝﾆﾁﾊ
OK
■
```

MID$

働き	文字列の中から、指定した数だけ一連の文字を取り出します。(⇒LEFT$、RIGHT$)

文法 **MID$(文字列，開始位置,n)**

文字列…最大文字数31文字
開始位置…文字列の最初の文字の位置を1とした取り出しを開始する文字の位置
n…………開始位置で指定した文字から取り出す文字の数 ∅～255

省略形 MI.

説明 文字列の中の開始位置から取り出されたn個の文字列が、この関数の値になります。開始位置が文字列の数より大きいときには、ヌルストリングが関数の値になります。

サンプルプログラム

```
10 REM * MID$ *
20 A$="オハヨウ"
30 FOR I=1 TO 4
40 PRINT MID$(A$,I,1)
50 NEXT
RUN
オ
ハ
ヨ
ウ
OK
■
```

LEN

働き 文字列の文字数を与えます。

文法 **LEN(文字列)**

省略形 LE.

説明 文字列に含まれるすべての文字数が、この関数の値になります。文字の数は∅～31で、文字列がヌルストリングのときは∅になります。なお、空白やコントロールコードなどの画面表示されない文字も1文字として数えます。

サンプルプログラム

```
10 REM * LEN *
20 INPUT"モジヲニュウリョクシテクダサイ";A$
30 L=LEN(A$)
40 PRINT"モジレツノ ナガサハ";L;"デス"
```

特殊関数

PEEK

働き 指定したメモリアドレスからデータを取り出します。(⇔POKE)

文法 **PEEK(アドレス)**

アドレス…メモリ内のアドレス
104ページのメモリマップをご覧ください。

省略形 PE.

説明 メモリ内の指定したアドレスから取り出した1バイトのデータが、この関数の値になります。

サンプルプログラム

☞POKE参照

POS

働き 画面のカーソルの水平位置を与える。

文法 **POS(数式)**

数式…ダミーの値∅

省略形 ありません。

説明 画面上の現在のカーソルの水平位置の値が、この関数の値になります。値の範囲は、∅～27です。

サンプルプログラム

```
10 REM * POS *
20 CLS
30 FOR X=0 TO 25
40 LOCATE X,0
50 PRINT POS(0)
60 PAUSE 10
70 NEXT
```

行番号30～70番でカーソルを水平位置のから25まで移動させ、その時の水平位置を行番号50番で順次表示します。

FRE

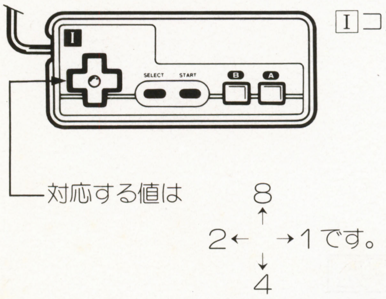
働き	ユーザーメモリの未使用領域のサイズを与えます。
文法	FRE
省略形	FR.
説明	BASICのプログラムで使っていないユーザーメモリのバイト数が、この関数の値になります。 注）BASICのバージョン、変数の状態、プログラムの有無、プログラムの実行前後によって値が違ってきます。

サンプルプログラム

```
PRINT FRE
 1854
OK
■
```

STICK

働き	コントローラの✜ボタンからの入力値を与えます。
文法	STICK(x) x…0, 1（0はⅠコントローラ、1はⅡコントローラに対応）
省略形	STI.
説明	コントローラの✜ボタンの値を与えます。



何も押さない場合は、0が与えられます。

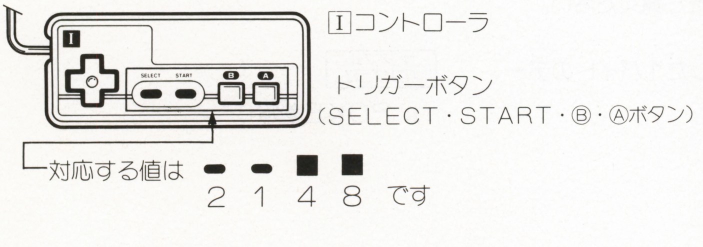
サンプルプログラム

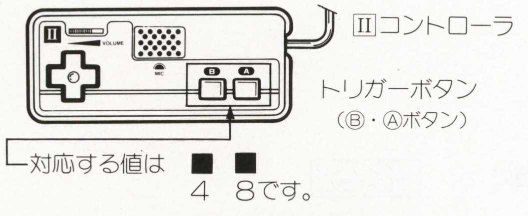
```
10 REM * STICK *
20 S=STICK(0)
30 IF S=0 THEN PRINT"ｵｼﾃｲﾅｲ"
40 IF S=1 THEN PRINT"ﾐｷﾞ"
50 IF S=2 THEN PRINT"ﾋﾀﾞﾘ"
60 IF S=4 THEN PRINT"ｼﾀ"
70 IF S=8 THEN PRINT"ｳｴ"
80 GOTO 20
```

RUNすると画面に連続的にオシテイナイが表示されますから、Ⅰコントローラの✜ボタンを押し、入力値が正しいかどうか確かめてください。
止めるときは、STOPキーを押してください。

STRIG

働き	コントローラのトリガーボタンからの入力状態値を与えます。
文法	STRIG(x) x…0, 1（0はⅠコントローラ、1はⅡコントローラに対応）
省略形	STR
説明	コントローラのトリガーボタンを押したときの値を与えます。





何も押さない場合は、0が与えられます。

サンプルプログラム

```
10 REM * STRIG *
20 T=STRIG(0)
30 IF T=1 THEN PRINT"ｽﾀｰﾄ"
40 IF T=2 THEN PRINT"ｾﾚｸﾄ"
50 IF T=4 THEN PRINT"B"
60 IF T=8 THEN PRINT"A"
70 GOTO 20
```

Ⅰコントローラのトリガーボタンを押して入力値が正しいかどうかを確かめてください。止めるときはSTOPキーを押してください。

CSRLIN

働き	カーソルの垂直位置を与えます。
文法	CSRLIN 引数はありません
省略形	CSR.
説明	現在のカーソルの垂直位置が、この変数の値になります。 値の範囲は0～23です。

サンプルプログラム

```
10 REM * CSRLIN *
20 CLS
30 FOR I=0 TO 20
40 LOCATE I,I
50 PRINT POS(0);",";CSRLIN
60 PAUSE 20
70 NEXT
```

行番号30～70番で、カーソルを水平、垂直とも1文字1行ずつ移動させ、その時のカーソルの水平、垂直位置を行番号50番で順次表示します。

入出力用文字関数

INKEY$

働き	キーボードからの一文字を与えます。
文法	INKEY$ {(n) / 省略} n＝0……引数
省略形	INK.
説明	キーボードから入力された1文字が、この関数の値になります。 (1)引数が省略された場合 キーが押されている時は、その文字が関数の値になります。 押されていなければ、ヌルストリングが関数の値になります。 (2)n＝0の場合 カーソルをブリンクし、1文字の入力があるまで待ちます。

サンプルプログラム

☞END、GOTO文参照

スプライト制御ステートメント

DEF SPRITE

働き　スプライト面上に表示したいスプライト(アニメキャラクタ)を定義します。

文法　DEF SPRITE n,(A,B,C,D,E)=文字式

文字式…CHR$(N)、または文字列"@ABC"、または文字変数

n…スプライト番号　0~7

A…配色番号　0~3(113ページのカラーチャート及びCGSETをご覧ください)

B…キャラクタ合成型　0~1〔0…8×8ドット(1文字の大きさ)、1…16×16ドット(4文字の大きさ)〕

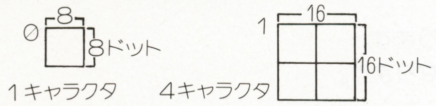
C…表示優先度　0~1　(0…バックグラウンド面より前のスプライト面を利用する、1…バックグラウンド面よりうしろのスプライト面を利用する)

D…X軸方向反転指示　0~1(0…キャラクタテーブルと同じ形、1…左右逆向のキャラクタとして定義する)

E…Y軸方向反転指示　0~1(0…キャラクタテーブルと同じ形、1…上下逆方向のキャラクタとして定義する)

省略形　DE.SP.

説明　スプライト面に表示したいアニメキャラクタを定義します。スプライト番号は、0~7の8コまで指定し、定義できます。キャラクタ合成型に0を指定すると、1キャラクタ(1文字の大きさ)でアニメキャラクタを構成します。1を指定すると、4キャラクタ(4文字の大きさ)アニメキャラクタを構成します。



表示優先度に0を指定すると、バックグラウンド面より手前のスプライト面上にアニメキャラクタを表示します。1を指定すると、バックグラウンド面の後方のスプライト面上にアニメキャラクタを表示します。

X軸方向・Y軸方向の反転指示に1を指定すると、それぞれの軸に対して反転したキャラクタで定義されます。0を指定すると反転しないキャラクタ(キャラクタテーブルA,Bの形)のままです。

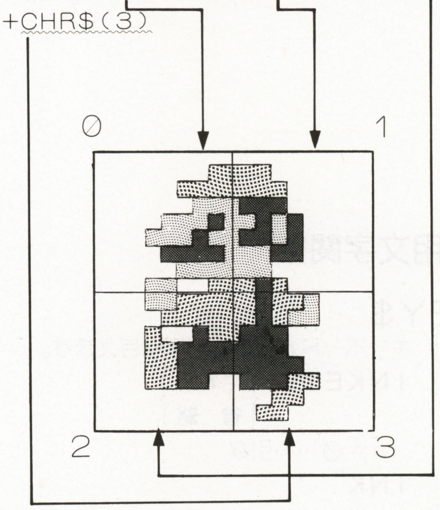
- MOVEコマンド-DEF MOVEとは独立して使用できます。
- CGEN命令と組み合わせて使用すれば、文字や記号をスプライトとして利用することができます。
- 文字例のCHR$(N)のNには、裏表紙のキャラクタテーブルの4すみに表示している数字やキャラクタテーブルA、Bのコードで指定します。数値は16進数&H○○も使用できます。
- CGEN2の場合は、DEF SPRITE 0,(0,1,0,0,0)=CHR$(68)+CHR$(69)+CHR$(70)+CHR$(71)は、DEF SPRITE 0,(0,1,0,0,0)="DEFG"とも指定できます。

サンプルプログラム

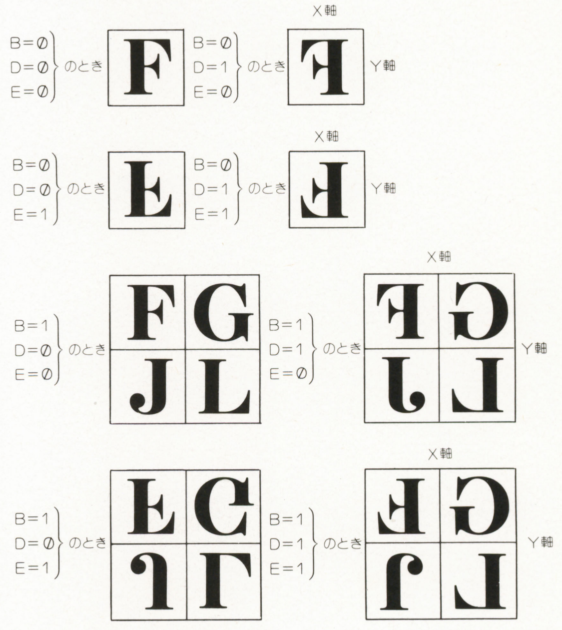
```
10 REM * DEF SPRITE *
20 SPRITE ON
30 DEF SPRITE 0,(0,1,0,0,0)=CHR$(0)
   +CHR$(1)+CHR$(2)+CHR$(3)
40 DEF SPRITE 1,(0,1,0,0,0)="@ABC"
50 SPRITE 0,100,150
60 SPRITE 1,100,100
```

〈文字式とキャラクタの対応〉

DEF SPRITE 0,(0,1,0,0,0)
=CHR$(0)+CHR$(1)+CHR$(2)
+CHR$(3)



〈反転キャラクタの定義〉



SPRITE

働き　定義されているスプライトを任意の位置に表示または消去します。

文法　SPRITE n[,x,y]

n…スプライト番号　0~7

x…スプライト面水平方向の座標　0~255

y…スプライト面垂直方向の座標　0~255

（ただしスプライト面上有効な範囲はX:0~240, Y:5~220）

省略形　SP.

説明　すでにDEF SPRITEで定義されているアニメキャラクタをスプライト面上の任意の位置に表示します。

水平方向・垂直方向の座標を省略すると、表示されているアニメキャラクタの中から、指定のスプライト番号のアニメキャラクタのみを消去します。〔SPRITE n〕

同一座標に複数のアニメキャラクタを表示した場合、スプライト番号の小さいものが手前に表示されます。（スプライト間の表示優先順位度が大きい）

水平方向には、同時に4つ（8文字分）までのアニメキャラクタが表示されます。（4つ以上のキャラクタは、画面上には見えなくなります）

アニメキャラクタのスプライト面への表示は、DEF SPRITEで定義したキャラクタの左上端の位置が指定座標にくるように表示されます。

BG GRAPHICと重ね合わせの場合は、スプライト面とBG GRAPHIC面の位置関係及びスプライトの表示座標の関係を考慮してください。

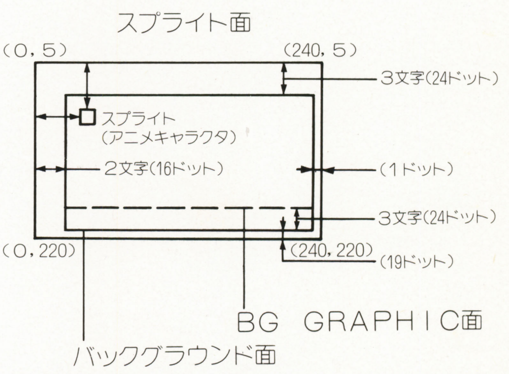
関係式は
$$\begin{cases} x=(X\times 8)+16 \\ y=(Y\times 8)+24 \end{cases}$$

x…スプライトのx座標

y…スプライトのy座標

X…BG GRAPHICのX座標

Y…BG GRAPHICのY座標



※画面の有効表示範囲はテレビによって多少異なります。

SPRITE ON

働き　スプライト面の表示を可能にします。スプライト表示モード。

文法　SPRITE ON

省略形　SP.O.

説明　スプライト面をバックグラウンド面に重ねて表示できる状態にします。

スプライト面上にあるスプライト（アニメキャラクタ）が、画面上に表示（見えるように）されます。

DEF SPRITE、またはDEF MOVEを実行する前には、必ずSPRITE ONが必要です。

SPRITE OFFが実行されるまで、継続して有効です。

サンプルプログラム

```
SPRITE ON
OK
■
 10 SPRITE ON
SPRITE OFF
OK
■
 10 SPRITE OFF
```

SPRITE OFF

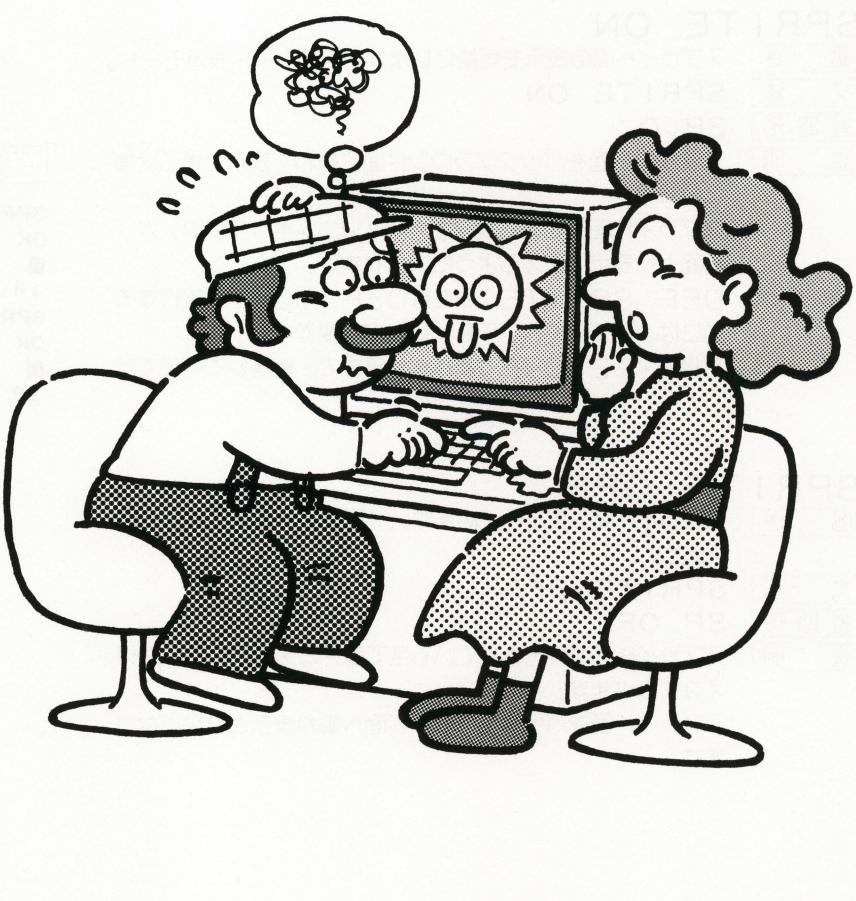
働き　スプライト面の表示を止めます。スプライト表示モードの取り消し。

文法　SPRITE OFF

省略形　SP.OF.

説明　スプライト面に表示されている全てのアニメキャラクタが見えなくなります。

スプライト面をバックグラウンド面へ重ね表示されなくなります。



ゲーム プログラム編

ファミリーベーシックで、オリジナルゲームを楽しもう。

サンプルプログラム

1、KNIGHT
2、SUPER MEMORY
3、UFO
4、ROUTE 66
5、TYPE MASTER
6、TURTLE
7、CARD

ファミリーベーシックの特長をフルに活用した、おもしろBASICプログラムが、次のページから記載してあります。さあ、プログラムを入力して楽しんでください。
サンプルプログラムに手を加えたり、ヒントにしながらオリジナルゲームを作れば、ファミリーベーシックの世界がいっそう拡がります。

サンプルプログラムのみかた

プログラムリスト

プログラムを入力する際、行番号や命令語を正しく入力してください。

```
5 CLS
10 SPRITE ON
20 CGSET 1,1
30 FOR N=0 TO 7
40 DEF MOVE(N)=SPRITE(0,N+1,3,255,0,0)
50 NEXT
60 MOVE 0,1,2,3,4,5,6,7
70 GOTO 60
80 END
```

長い命令文の場合、プログラムリストと実際の入力表示画面は異なります。行が変わっても続けて入力してください。

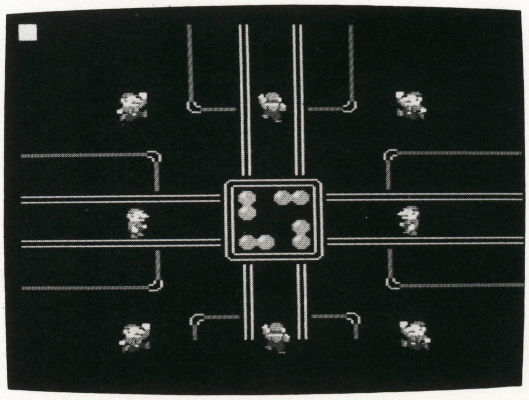
（例）

```
30 FOR N=0 TO 7
40 DEF MOVE(N)=SPRITE(0,N+1,
3,255,0,0)
50 NEXT
```

［入力はリストの通り打ち込んでください。｛GOTOやGOSUBまたは、IF～THENの後に続く行番号はスペースを入れないでください。メモリオーバー（OM ERROR）となる場合があります。
（例　GOTO120）｝］

〈サンプル画面〉

BG　GRAPHICで背景画を描こう。アニメキャラクタが動きまわるのは、どんなところかな？海、山、それとも宇宙？　こんどはBASICでオリジナルゲームをプログラミング。MOVEコマンドがあるから、ドキドキハラハラのゲームがかんたんにつくれるよ。VIEW命令で、背景画を重ねたら、楽しいゲームの完成！



★プログラムの変更や修正をする時の注意

- BASICプログラムの作成や変更および修正は、必ずBG　GRAPHIC（背景）画面を消してから行なってください。背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG　GRAPHIC画面を消すには［SHIFT］キーを押しながら［CLR HOME］キーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

背景画のデータ

BG　GRAPHICでは、ヨコ28マス×タテ21行の範囲内で背景の線が描けます。このデータは、BG　GRAPHICで描く各キャラクタのデザインデータを示します。

「背景画のデータ」で、例えば、

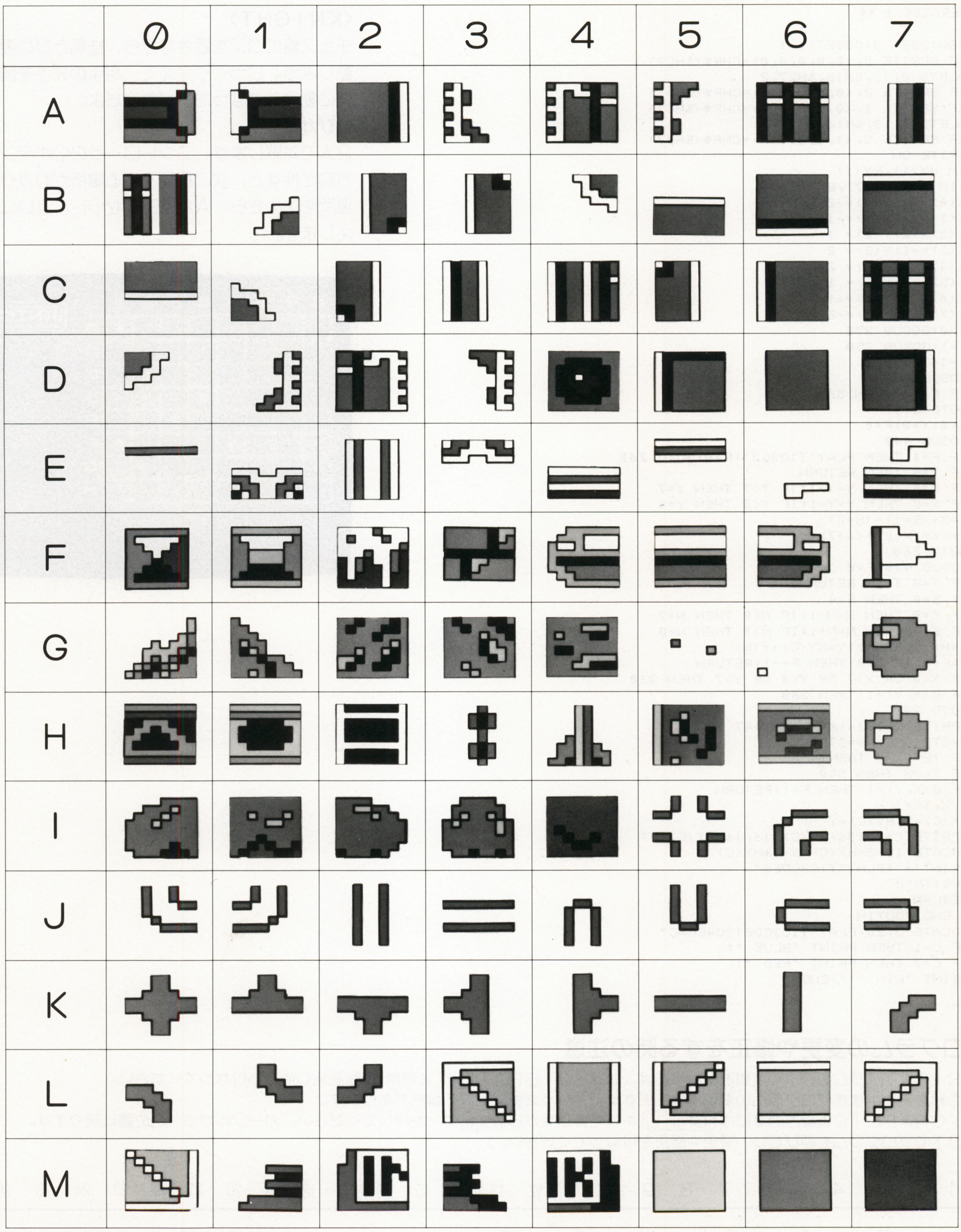
■K50とあれば、Kグループの5のキャラクタを使い、配色は0（SELECT　モードで選択する0番の配色）を意味します。

■Bグループの7のキャラクタを使い、配色は3を意味します。

※数字や記号で表示されている「背景画のデータ」（例：数字の1や英数字のAなど）は、CHARモードで直接キーボードから入力します。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0										K60			J20			J20			K60									
1										K60			J20			J20			K60									
2										K60			J20			J20			K60									
3										K60			J20			J20			K60									
4										K60			J20			J20			K60									
5										J00	K50	K50	J20			J20	K50	K50	J10									
6													J20			J20												
7													J20			J20												
8	K50	K50	K50	K50	K50	K50	K50	I70					J20			J20					I60	K50	K50	K50	K50	K50	K50	K50
9								K60					J20			J20					K60							
10								K60				I60	J30	J30	J30	J30	I70				K60							
11	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J20	G72		G72	G72	J20	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30
12												J20	G72				J20											
13												J20				G72	J20											
14	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J20	G72	G72		G72	J20	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30
15								K60				J00	J30	J30	J30	J30	J10				K60							
16								K60					J20			J20					K60							
17	K50	K50	K50	K50	K50	K50	K50	J10					J20			J20					J00	K50	K50	K50	K50	K50	K50	K50
18													J20			J20												
19										I60	K50	K50	J20			J20	K50	K50	I70									
20										K60			J20			J20			K60									

BG GRAPHICで使えるキャラクタは、下図の13種類のグループ(A~M)があり、配色もそれぞれ0~3の4種から選べます。



サンプルプログラム1

```
** EXERCISE 1 **

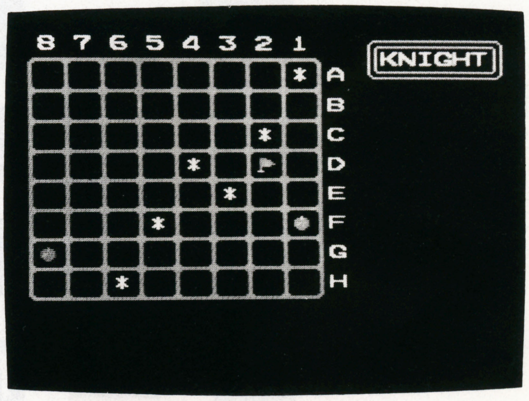
10 VIEW:CGEN 3:CGSET 1,1
20 DEF SPRITE 0,(0,0,0,0,0)=CHR$(&HC7)
30 PALETS 0,13,&H16,&H27,2
40 DEF SPRITE 2,(0,0,0,0,0)=CHR$(&HD7)
50 DEF SPRITE 1,(0,0,0,0,0)=CHR$(&HC7)
60 PALETS 1,13,&H16,&H17,4
70 DEF SPRITE 3,(1,0,0,0,0)=CHR$(&HD7)
80 SPRITE ON
90 DIM HX(1),HY(1)
100 DIM X(7),Y(7),B(7,7)
110 X(0)=-1:Y(0)=-2
120 X(1)=-2:Y(1)=-1
130 X(2)=-2:Y(2)= 1
140 X(3)=-1:Y(3)= 2
150 X(4)= 1:Y(4)= 2
160 X(5)= 2:Y(5)= 1
170 X(6)= 2:Y(6)=-1
180 X(7)= 1:Y(7)=-2
190 C=0:GOSUB 250
200 C=1:GOSUB 250
210 C=1+(C=1)
220 GOSUB 390
230 IF F=-1 THEN 560
240 GOTO 210
250 X=0:Y=0:F=0
260 GOSUB 440
270 IF F=1 THEN PLAY"T103C2":F=0:GOTO 260
280 IF T=8 THEN RETURN
290 IF S=4 THEN Y=Y+1:IF Y>7 THEN Y=7
300 IF S=8 THEN Y=Y-1:IF Y<0 THEN Y=0
310 X=X+(S=1)-(S=2)
320 X=-X*(X>0)+(X>7)
330 GOTO 260
340 GOSUB 440:F=0
350 IF T=8 THEN RETURN
360 IF S=0 THEN S=4
370 IF S=8 THEN N=N-1:IF N<0 THEN N=7
380 IF S=4 THEN N=N+1:IF N>7 THEN N=0
390 X=HX(C)+X(N):Y=HY(C)+Y(N)
400 F=F+1:IF F>8 THEN F=-1:RETURN
410 IF X<0 OR X>7 OR Y<0 OR Y>7 THEN 360
420 IF B(X,Y)=1 THEN 360
430 GOTO 340
440 SPRITE C,136-16*X,16*Y+47
450 T=STRIG(C):S=STICK(C)
460 IF (S+T)=0 THEN 450
470 IF T<>8 THEN 540
480 IF B(X,Y)=1 THEN F=1:RETURN
490 B(X,Y)=1
500 HX(C)=X:HY(C)=Y
510 SPRITE C+2,136-HX(C)*16,16*HY(C)+47
520 LOCATE 15-2*HX(C),3+2*HY(C)
530 PRINT"*":PLAY"T103CDEG"
540 SPRITE C
550 RETURN
560 ' END ROUTIN
570 LOCATE 3,20:PLAY"T103CDET204EGAC"
580 IF C=1 THEN PRINT "BLUE ";
590 IF C=0 THEN PRINT "RED ";
600 PRINT "WIN !!":END
```

〈KNIGHT〉

チェス盤にコマを置きながら、桂馬とびの要領で移動しよう。じっくり考えて、相手の先手を読もう。先に動けなくなった方が負けだよ。

遊びかた

2人で対戦します。コントローラの✚ボタンの上下方向を押すと、次に移動できる場所がわかります。場所を決めたら、Aボタンでセット。互いにくりかえします。



★プログラムの変更や修正をする時の注意

- BASICプログラムの作成や変更および修正は、必ずBG GRAPHIC(背景)画面を消してから行なってください。
  背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG GRAPHIC画面を消すには[SHIFT]キーを押しながら[CLR HOME]キーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1		8		7		6		5		4		3		2		1				I60	J30	J30	J30	J30	J30	J30	J70	
2	K72	K52	K22	K52	K22	K52	K22	K52	K22	K52	K22	K52	K22	K52	K22	K52	L02			J20	K	N	I	G	H	T	J20	
3	K62		K62		K62		K62		K62		K62		K62		K62		K62	A		J00	J30	J30	J30	J30	J30	J30	J10	
4	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
5	K62		K62		K62		K62		K62		K62		K62		K62		K62	B										
6	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
7	K62		K62		K62		K62		K62		K62		K62		K62		K62	C										
8	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
9	K62		K62		K62		K62		K62		K62		K62		K62		K62	D										
10	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
11	K62		K62		K62		K62		K62		K62		K62		K62		K62	E										
12	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
13	K62		K62		K62		K62		K62		K62		K62		K62		K62	F										
14	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
15	K62		K62		K62		K62		K62		K62		K62		K62		K62	G										
16	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
17	K62		K62		K62		K62		K62		K62		K62		K62		K62	H										
18	L12	K52	K12	K52	K12	K52	K12	K52	K12	K52	K12	K52	K12	K52	K12	K52	L22											
19																												
20																												

サンプルプログラム2

```
** EXERCISE 2 **

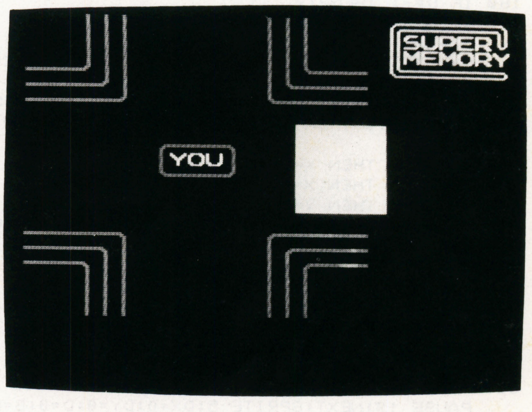
10 VIEW:CGEN 2
20 MAX=5:I=Z:X=Y:C=0
30 PP$="CFGE"
40 Z$=CHR$(254):Z$=Z$+Z$+Z$+Z$+Z$
50 N$="      "
60 DIM PL(MAX),PX(3),PY(3),C(3)
70 PX(0)=16:PY(0)=8:C(0)=2
80 PX(1)=0:PY(1)=8:C(1)=4
90 PX(2)=8:PY(2)=0:C(2)=6
100 PX(3)=8:PY(3)=16:C(3)=8
110 '
120 PL(Z)=RND(4)
130 FOR I=0 TO Z
140 X=PX(PL(I)):Y=PY(PL(I))
150 C=C(PL(I))
160 PALETB 0,13,13,13,13
170 GOSUB 440
180 PLAY"T204"+MID$(PP$,PL(I)+1,1)+"3"
190 GOSUB 500
200 PALETB 0,13,&H16,&H27,2
210 PAUSE 10
220 NEXT
230 '
240 I=0
250 LOCATE 9,10:PRINT"YOU"
260 A$=INKEY$:IF A$="" THEN 260
270 IF A$<CHR$(28) OR A$>CHR$(31) THEN PLAY"T101C1C1C1":GOTO 260
280 IF (ASC(A$)-28)<>PL(I) THEN PLAY"T205C2R2F2R2E2":LOCATE 9,10:
    PRINT"   ":GOTO 130
290 PALETB 0,13,13,13,13
300 X=PX(PL(I)):Y=PY(PL(I))
310 C=C(PL(I))
320 GOSUB 440
330 PLAY"T204"+MID$(PP$,PL(I)+1,1)+"3"
340 GOSUB 500
350 PALETB 0,13,&H16,&H27,2
360 I=I+1:IF I<=Z THEN 250
370 Z=Z+1:IF Z>MAX THEN 410
380 LOCATE 9,10:PRINT"   "
390 PAUSE 50
400 GOTO 110
410 ' END
420 CLS:CGSET 1,1:PRINT"GOOD!!"
430 END
440 FOR J=0 TO 5
450 LOCATE X,Y+J
460 PRINT Z$
470 NEXT
480 PALETB 0,13,C,C,C
490 RETURN
500 PALETB 0,13,13,13,13
510 FOR J=0 TO 5
520 LOCATE X,Y+J
530 PRINT N$
540 NEXT
550 RETURN
```

〈SUPER　MEMORY〉

コンピュータが君の記憶力をテスト。上下左右、カラーパネルの点滅順序を君は正確に記憶できるかな?

遊びかた

１人で遊びます。まず、上下左右の4つの方向にカラーパネルがランダムに点滅しますから、順序を憶えておいてください。中央に、"YOU" とでてきたら同じ順序でカーソルキーを押します。画面の上下左右が［▲］［▼］［◀］［▶］に対応しています。途中でまちがえると、コンピュータがブーイング。もう一度、同じ順序で点滅します。行番号10で点滅回数の最大値を定めています。数値を大きくすれば点滅回数がふえます。



★プログラムの変更や修正をする時の注意

- BASICプログラムの作成や変更および修正は、必ずBG　GRAPHIC(背景)画面を消してから行なってください。
  背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG　GRAPHIC画面を消すには［SHIFT］キーを押しながら［CLR/HOME］キーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1					K62	K62	K62								K62	K62	K62					I60	J30	J30	J30	J30	J30	I70
2					K62	K62	K62								K62	K62	K62					J20	S	U	P	E	R	J50
3					K62	K62	K62								K62	K62	K62					J20	M	E	M	O	R	Y
4		K52	K52	K52	L22	K62	K62								K62	K62	L12	K52	K52	K52		J00	J30	J30	J30	J30	J30	J70
5		K52	K52	K52	K52	L22	K62								K62	L12	K52	K52	K52	K52								
6		K52	K52	K52	K52	K52	L22								L12	K52	K52	K52	K52	K52								
7																												
8																												
9									K72	K52	K52	K52	L02															
10									K62				K62															
11									L12	K52	K52	K52	L22															
12																												
13																												
14																												
15		K52	K52	K52	K52	K52	L02								K72	K52	K52	K52	K52	K52								
16		K52	K52	K52	K52	L02	K62								K62	K72	K52	K52	K52	K52								
17		K52	K52	K52	L02	K62	K62								K62	K62	K72	K52	K52	K52								
18					K62	K62	K62								K62	K62	K62											
19					K62	K62	K62								K62	K62	K62											
20					K62	K62	K62								K62	K62	K62											

サンプルプログラム3

```
** EXERCISE 3 **

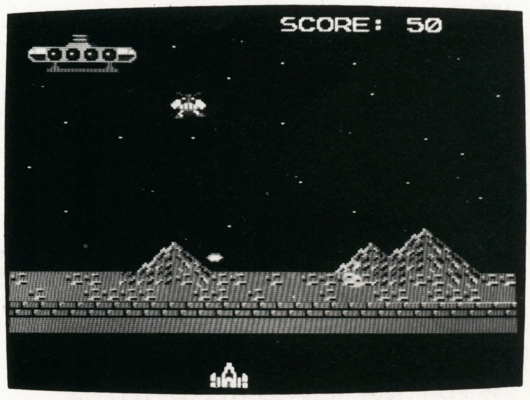
10 VIEW:CGEN 2:PLAY"T1C1"
20 U0$="89:;":U1$="<=>?":B$="ピプペポ":FOR I=0 TO 3:
   X$=X$+CHR$(180+I):NEXT:V=1
30 LG=16:RG=220:TG=50:BG=100:GX=0:GY=0:GV=16:BX=100:
   BY=220:BV=6:DX=0:DY=0:DV=30:R=12:X=0:Y=0:D=0
40 G0$=CHR$(212):G1$=CHR$(213):D0$=CHR$(209):D1$=G1$
50 PALETS 1,13,48,22,7:PALETS 2,13,48,22,1:PALETS 3,13,48,22,1:
   DEF SPRITE 1,(2,1,0,0,0)=X$:DEF SPRITE 6,(3,1,0,0,0)=B$:
   GOSUB310:SPRITE ON
60 VX=V+RND(R):VY=V+RND(R)
70 C=RND(5):Y=Y+VY:X=X+VX*SGN(C):GOSUB260:T=STRIG(0):S=STICK(0)
80 SWAP G0$,G1$:SWAP D0$,D1$:SWAP U0$,U1$:GOSUB310
90 IF G=-1 THEN120
100 IF T<8 THEN160
110 GX=BX+6:GY=BY:G=-1
120 GY=GY-GV
130 IF GY<TG THEN G=0:GX=0:GY=0:GOTO160
140 SPRITE 7,GX,GY
150 IF ABS(GX-X-5)<6 THEN IF ABS(GY-Y+8)<8 THEN330
160 IF S=1 THEN BX=BX+BV:IF BX>RG THEN BX=RG
170 IF S=2 THEN BX=BX-BV:IF BX<LG THEN BX=LG
180 IF D=-1 THEN210
190 IF ABS(BX-X)>5 THEN240
200 DX=X+6:DY=Y:D=-1
210 DY=DY+DV:IF DY>255 THEN SPRITE 5:D=0:DX=0:DY=0:GOTO240
220 SPRITE 5,DX,DY
230 IF (BY-DY)<6 THEN IF ABS(BX-DX+6)<8 THEN350
240 IF BG<210 THEN IF RND(4)=0 THEN TG=TG+V:BG=BG+V
250 GOTO70
260 IF X<LG THEN X=LG:VX=-VX
270 IF X>RG THEN X=RG:VX=-VX
280 IF Y<TG THEN Y=TG:VY=-VY
290 IF Y>BG THEN Y=BG:VY=-VY:IF BG>=210 THEN360
300 RETURN
310 DEF SPRITE 0,(1,1,0,0,0)=U0$:DEF SPRITE 5,(0,0,0,0,0)=D0$:
    DEF SPRITE 7,(3,0,0,0,0)=G0$
320 SPRITE 0,X,Y:SPRITE 7,GX,GY:SPRITE 5,DX,DY:SPRITE 6,BX,BY:RETURN
330 FOR I=0 TO 9:SWAP X$,U0$:DEF SPRITE 0,(3,1,0,0,0)=U0$:
    SPRITE 0,X,Y:PAUSE 10:PLAY"O5F1":NEXT
340 PR=PR+10:LOCATE 15,0:PRINT"SCORE:";PR:GX=0:GY=0:V=1+PR/20:GOTO30
350 FOR I=0 TO 5:SPRITE 6:PLAY"O1B1":PAUSE 8:SPRITE 6,BX,BY:
    PAUSE 10:NEXT:SPRITE 5:DX=0:DY=0:D=0:B=B+1:IF B<4 THEN30
360 PLAY"O1C4EC":INPUT"RETURN";I:RUN
```

〈UFO〉

謎の飛行物体からファイターフライ出現！君のスターシップがねらわれている。ファイターフライの攻撃からすばやく身をかわし、さあ反撃開始だ。

遊びかた

１人で遊びます。Ⅰコントローラの✚ボタンの左右方向を押して、スターシップを動かします。ボタンAを押せばミサイル発射。うまくファイターフライに命中させてください。



★プログラムの変更や修正をする時の注意

- ●ＢＡＳＩＣプログラムの作成や変更および修正は、必ずＢＧ　ＧＲＡＰＨＩＣ(背景)画面を消してから行なってください。
  背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- ●ＢＧ　ＧＲＡＰＨＩＣ画面を消すには SHIFT キーを押しながら CLR HOME キーを押してください。カーソルがホーム位置に戻ります。
- ●ＬＩＳＴでプログラムを呼びだし、変更や修正を行なってください。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1				M11	M31					G62			G52															
2		F41	D41	D41	D41	D41	F61									G62				G62					G52			
3			K12	K52	K52	K12							G62															
4		G52																			G62							G62
5																												
6				G62												G62												
7							G62				G62								G52					G62				
8																										G62		
9		G62																										
10																G52	G62						G62		G52			
11								G52																			G62	
12				G62																G62								
13													G62															
14																							G00	G10				
15									G00	G10										G00	G10	G00	G20	G30	G10			
16								G00	G20	G30	G10								G00	G20	G20	G20	G20	G30	G30	G10		
17	G40	M70	M70	G40	G40	M70	G20	G20	G20	G40	G30	G40	G40	M70	G40	M70	G40	G40	G40	G40	G20	G20	G40	G40	G30	G30	G40	G40
18	M70	G40	M70	M70	G40	G40	G40	G40	M70	M70	G40	G40	M70	G40	M70	M70	M70	G40	M70	M70	G40	G40	M70	G40	G40	G40	M70	M70
19	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
20	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72

サンプルプログラム4

```
** EXERCISE 4 **

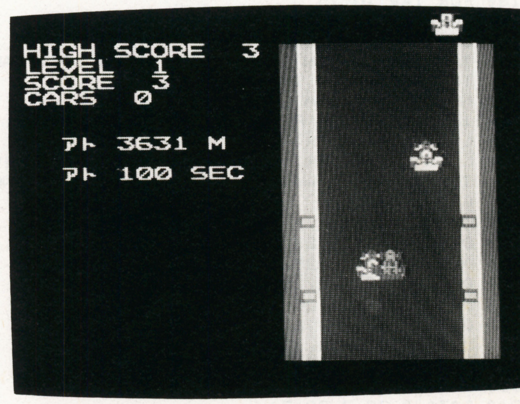
10 VIEW:CGEN 2:CGSET 1,2:PALETS 0,13,22,39,2:SPRITE ON:PLAY"T103"
20 X1=150:X2=170:X3=190:X6=136:X7=208:
   C0$="ﾙﾚﾛﾜ":C1$="ﾝｦｧｨ":B$=CHR$(214)
30 FOR I=4 TO 7:DEF SPRITE I,(0,0,0,0,0)=B$:NEXT
40 F=2:SC=0:CA=3:L=0:X=208:Y0=150:V=0
50 L=L+1:M=L*3+15:MV=M+33:D=(M+27)*400
60 C=0
70 S=STICK(0):T=STRIG(0):V=V-1
80 IF S=1 THEN X=X+5
90 IF S=2 THEN X=X-5
100 IF T=8 THEN V=V+4:IF V>MV THEN V=MV
110 IF T=4 THEN V=V-4
120 IF V<1 THEN V=0
130 IF X<143 OR X>193 THEN310
140 F=1
150 Y=Y0-V/2:DEF SPRITE 0,(0,1,0,0,0)=C0$:SPRITE 0,X,Y:SWAP C0$,C1$
160 IF F1=-1 AND F2=-1 AND F3=-1 THEN SC=SC+1:F1=0:F2=0:F3=0
170 FOR I=1 TO 3:DEF SPRITE I,(1,1,0,0,0)=C0$:NEXT
180 YY=(40-Y5)*(Y5>40):SPRITE1,X1,Y1:SPRITE2,X2,Y2:SPRITE3,X3,Y3:
 SPRITE4,X6,Y5:SPRITE5,X7,Y5:SPRITE6,X6,YY:SFRITE7,X7,YY
190 IF ABS(Y1-Y)<12 THEN F1=V>V1:IF ABS(X1-X)<10 THEN340
200 IF ABS(Y2-Y)<12 THEN F2=V>V2:IF ABS(X2-X)<10 THEN340
210 IF ABS(Y3-Y)<15 THEN F3=V>V3:IF ABS(X3-X)<10 THEN340
220 Y1=Y1+V-V1:Y2=Y2+V-V2:Y3=Y3+V-V3:Y5=Y5+V:XX=XX+2:X1=ABS(XX)+143:
    IF XX>48 THEN XX=-51
230 IF Y1>250 THEN Y1=0:V1=RND(M)+30
240 IF Y2>250 THEN Y2=0:X2=RND(58)+138:V2=RND(M)+30
250 IF Y3>250 THEN Y3=0:X3=RND(58)+138:V3=RND(M)+30
260 Y5=-Y5*(Y5<250):Y1=Y1-(Y1<0)*255:Y2=Y2-(Y2<0)*255:Y3=Y3-(Y3<0)*255
270 D=D-V:IF C-C/100*100=0 THEN GOSUB380
280 IF D<0 THEN GOSUB380:PRINT"GOOD!  ":PLAY"CGDAB":SC=SC+(500-C)/10:
    GOTO50
290 C=C+1:IF C<401 THEN70
300 PRINT"TIME UP":PLAY"BAGFEDC":F=3:L=L-1:GOTO340
310 V=V-4:IF F=2 THEN V=V+4:IF V>MV-15 THEN V=MV-15
320 IF V<0 THEN V=0:F=2
330 GOTO150
340 PLAY"BAEDC":CA=CA-1:IF CA>-1 THEN NX=208:V=0:GOSUB380:
    ON F GOTO 70,70,50
350 PRINT"GAME OVER!":PRINT"TRY AGAIN"
360 IF STRIG(0)<>1 THEN360
370 GOSUB380:GOTO10
380 IF SC>HI THEN HI=SC
390 LOCATE 0,0:PRINT"HIGH SCORE ";HI:PRTINT"LEVEL ";L:PRINT"SCORE ";SC:
    PRINT"CARS ";CA:PRINT:PRINT:PRINT"  ｱﾄ";D/4;" M ":PRINT:
    PRINT"  ｱﾄ";400-C;" SEC ":PRINT:RETURN
```

〈ROUTE 66〉

はてしなくつづく直線ロードをレーシングカーでぶっとばせ！アクセルで加速、ハンドルをすばやく切り、進路をジャマするヤツを追い越そう。

遊びかた

１人で遊びます。指定された距離をタイムリミット内に走破してください。Ⅰコントローラの✣ボタンの左右方向を押してハンドルを操作、アクセルはAボタン、ブレーキはBボタンです。他の車に衝突すると、爆発します。スタート位置で加速してから、コースに入ってください。



※OM ERROR（オーバーメモリーエラー）が出た場合は、入力したプログラム中に必要のない空きスペースが含まれていることがあります。
余分なスペースを省略して、再度プログラムを実行してください。

★プログラムの変更や修正をする時の注意

- BASICプログラムの作成や変更および修正は、必ずBG GRAPHIC(背景)画面を消してから行なってください。
  背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG GRAPHIC画面を消すには［SHIFT］キーを押しながら［CLR HOME］キーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
1															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
2															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
3															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
4															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
5															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
6															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
7															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
8															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
9															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
10															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
11															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
12															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
13															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
14															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
15															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
16															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
17															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
18															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
19															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
20															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		

サンプルプログラム5

```
** EXERCISE 5 **

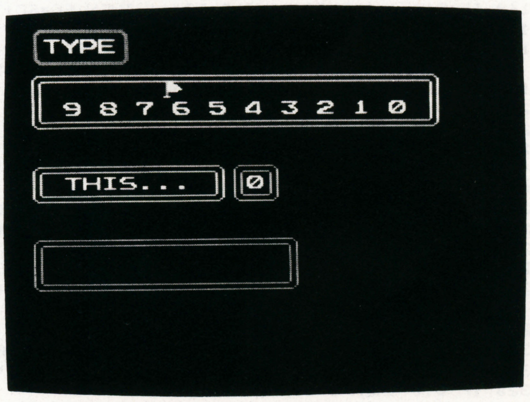
10 CGEN 3:CGSET 1,1
20 VIEW:PLAY"T104C1"
30 PO=PP:I=J:K=L
40 PO$="O102030405"
50 PP$="CDEFGAB"
60 A$=B$
70 M3$="ｶﾅﾃﾞｽﾖ!"
80 N$=CHR$(227):NN$=N$+N$+N$+N$+N$+N$+N$
90 DIM DP(64),PP(64)
100 FOR I=0 TO 63:READ DP(I):NEXT
110 DEF SPRITE 0,(0,0,0,0,0)=CHR$(199)
120 SPRITE ON
130 VIEW:K=0:LOCATE 2,5:FOR I=9 TO 0 STEP -1:PRINT I;:NEXT
140 PAUSE 100:A$=CHR$(48+RND(43)):SPRITE 0
150 LOCATE 3,10:PRINT "THIS...";
160 LOCATE 13,10:PRINT A$
170 GOSUB 460
180 FOR I=0 TO 75
190 FOR J=0 TO 5:B$=INKEY$
200 SPRITE 0,38+2*I,53
210 IF B$<>"" THEN SWAP A$,B$:GOSUB 370:SWAP A$,B$:
    IF A$=B$ THEN I=500:J=I
220 NEXT:NEXT:LOCATE 5,3:PRINT NN$
230 IF I>100 THEN PLAY"O4B1AG2FE3D4C":K=K+1:GOTO 140
240 LOCATE 4,15:PRINT K;"ｶｲ ｾｲｶｲ"
250 LOCATE 3,16:PRINT"TRY AGAIN ?";:A$=INKEY$(0):
    IF A$="N" OR A$="ﾔ" THEN LOCATE 0,18:END
260 GOTO 130
270 '
280 DATA 0,21,22,23,24,24,25,25,26,27
290 DATA 14,14,5,28,6,7
300 DATA 29,21,22,23,24,24,25,25,26,27
310 DATA 14,14,5,28,6,7,21
320 DATA 7,3,2,9,16,10,10
330 DATA 11,19,11,12,13,4,4
340 DATA 20,21,14,17,8,17,18
350 DATA 3,15,1,18,0
360 DATA 0,28,0,28,7
370 IF A$ <" " THEN PLAY"O1C1EC":A$=""
380 IF A$ >"_" THEN PLAY"O5B1AB":LOCATE 5,3:PRINT M3$:A$=" "
390 IF A$="" THEN RETURN
400 IF A$="F" OR A$="J" THEN PLAY"O0A1:O0B1":GOTO 450
410 PO=DP(ASC(A$)-32)/7
420 PP=DP(ASC(A$)-32)-PO*7
430 PLAY MID$(PO$,PO*2+1,2)+MID$(PP$,PP+1,1)
440 PLAY MID$(PP$,PP+1,1)
450 RETURN
460 FOR J=0 TO 5:GOSUB 400:NEXT:RETURN
```

〈TYPE MASTER〉

キーの配列ってなかなか覚えられないよね。でも、もう大丈夫。楽しみながらキーボードのキーの配列を覚えちゃおう、というのがこのタイプマスターなんだ。これで君も名タイピスト?!

遊びかた

1人で遊びます。中段の右のらんに表示された、英文字や数字、記号と同じものをキーボードのキーからさがしだして、入力してください。ハタが右に移動しおわると、タイムオーバーです。



★プログラムの変更や修正をする時の注意

- BASICプログラムの作成や変更および修正は、必ずBG GRAPHIC(背景)画面を消してから行なってください。
  背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG GRAPHIC画面を消すには SHIFT キーを押しながら CLR HOME キーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0		K72	K52	K52	K52	K52	L02																					
1		K62	T	Y	P	E	K62																					
2		L12	K52	K52	K52	K52	L22																					
3		I60	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	I70				
4		J20																						J20				
5		J20																						J20				
6		J00	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J10				
7																												
8																												
9		I60	J30	J30	J30	J30	J30	J30	J30	J30	J30	I70	I62	J32	I72													
10		J20										J20	J22		J22													
11		J00	J30	J30	J30	J30	J30	J30	J30	J30	J30	J10	J02	J32	J12													
12																												
13																												
14		I63	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	I73											
15		J23															J23											
16		J23															J23											
17		J03	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J13											
18																												
19																												
20																												

サンプルプログラム6

```
** EXERCISE 6 **

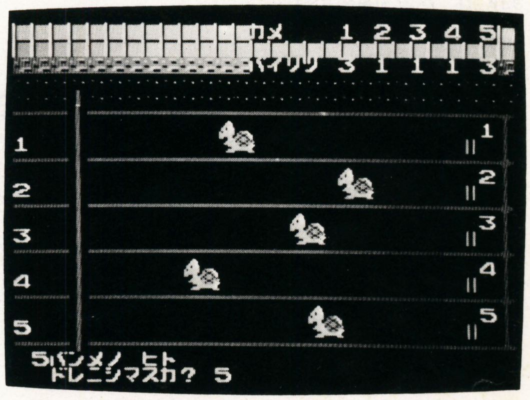
10 CLS:CGEN 2:CGSET 1,2
20 I=K:A=J:BT=5:MAX=3
30 PR=PP:W=-1
40 PLAY"O4T1"
50 P$="C2D2E2F2G2A2B2C2"
60 DIM AX(4),AY(4),Q(4)
70 DIM P(5,1),C(5)
80 INPUT "ﾅﾝﾆﾝ ﾃﾞｽｶ? ",PL
90 IF PL=0 OR PL>5 THEN PLAY"C1CC":CLS:GOTO 80
100 VIEW
110 LOCATE 13,0:PRINT"ｶﾒ    "
120 LOCATE 13,2:PRINT"ﾊﾞｲﾘﾂ"
130 FOR I=0 TO 4
140 Q(I)=RND(MAX)+BT
150 C(I)=MAX+BT-Q(I)
160 DEF SPRITE I,(0,1,0,0,0)=CHR$(184)+CHR$(185)
    +CHR$(186)+CHR$(187)
170 LOCATE 17+2*I,0:PRINT I+1
180 LOCATE 17+2*I,2:PRINT C(I)
190 AX(I)=220
200 P(I,0)=-1
210 NEXT
220 '
230 FOR I=0 TO PL-1
240 LOCATE 0,21:PRINT I+1;"ﾊﾞﾝﾒﾉ ﾋﾄ "
250 INPUT " ﾄﾞﾚﾆｼﾏｽｶ? ",M
260 IF M=0 OR M>5 THEN PLAY"C1CC":GOTO 240
270 P(I,0)=M-1
280 NEXT
290 FOR M=4 TO 20:LOCATE 3,M:PRINT CHR$(238):NEXT
300 FOR M=0 TO 4:LOCATE 25,6+3*M:PRINT M+1:NEXT
310 '
320 SPRITE ON
330 FOR I=0 TO 4
340 A=RND(RND(Q(I)))
350 AY(I)=71+24*I
360 AX(I)=AX(I)-A*2
370 IF AX(I)<0 THEN AX(I)=0
380 PLAY MID$(P$,A*2+1,2)
390 SPRITE I,AX(I),AY(I)
400 IF AX(I)<50 THEN GOSUB 560
410 NEXT
420 K=K+1
430 IF K=20 OR K=50 THEN GOSUB 450
440 GOTO 310
450 '
460 FOR I=0 TO 4
470 LOCATE 10,10:PRINT"CHANGE!"
480 A=RND(8)
490 IF A=7 THEN Q(I)=9:GOTO 530
500 IF A=6 THEN IF RND(5)=4 THEN Q(I)=0:GOTO 530
510 IF A<5 THEN 530
520 Q(I)=11-Q(I)
530 NEXT
540 LOCATE 10,10:PRINT"
550 RETURN
560 '
570 IF I=W THEN RETURN
580 IF W<>-1 THEN 600
590 W=I:RETURN
600 '
610 PLAY"T6CEDFEGC2T1"
620 LOCATE 10,6:PRINT"WIN";W+1;"-";I+1
630 SPRITE OFF
640 FOR J=0 TO PL-1
650 PR=0
660 IF P(J,0)=W THEN PR=2
670 IF P(J,0)=I THEN PR=1
680 PP=PR*C(P(J,0))
690 LOCATE 9,9+J*2:PRINT J+1;"ﾊﾞﾝ";PP;"ﾃﾝ"
700 P(J,1)=P(J,1)+PP
710 NEXT
720 LOCATE 16,22:PRINT"TRY AGAIN ?";:A$=
    INKEY$(0)
730 IF A$="N" OR A$="ﾔ" THEN 750
740 RUN
750 CLS:LOCATE 5,10:PRINT"----- END -----":
    PLAY"CDGAC":END
```

〈TURTLE〉

5匹のカメさんがゴールを目指して競争するよ。倍率の低いのが、本命なんだけど、意外な逆転もあるんだよ。ハラハラ、ドキドキ、勝負はゴールインまでわからない。

遊びかた

5人まで一緒に遊べます。まず何人で遊ぶかを決めたら、倍率を参考にして、何番のカメさんが1着になるかを予想します。ピタリと当たれば、その倍率で得点されます。実行中に表示される"CHANGE!"はゲームをおもしろくするために、内部で乱数の変更をしていることを示すメッセージです。



★プログラムの変更や修正をする時の注意

- BASICプログラムの作成や変更および修正は、必ずBG GRAPHIC(背景)画面を消してから行なってください。背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG GRAPHIC画面を消すには SHIFT キーを押しながら CLR HOME キーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50
1	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52
2	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42
3	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52
4	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52
5	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K22
6				J20																						J22		K62
7	1			J20																						J22		K62
8	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
9				J20																						J22		K62
10	2			J20																						J22		K62
11	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
12				J20																						J22		K62
13	3			J20																						J22		K62
14	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
15				J20																						J22		K62
16	4			J20																						J22		K62
17	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
18				J20																						J22		K62
19	5			J20																						J22		K62
20	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K12

サンプルプログラム7

```
** EXERCISE 7 **

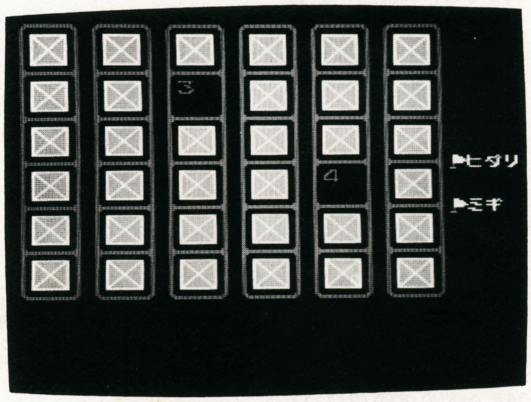
10 VIEW:CGEN 3:CGSET 1,1:SPRITE ON
20 I=J:A=C:S=T:X=-1:Y=-1:ED=P:CX=XY:T0=T1:TT=0
30 N$=CHR$(254):N$=N$+N$+N$+N$:M0$=CHR$(243)+CHR$(247):
   M1$=CHR$(245)+CHR$(248)
40 DIM D(5,5),PT(17),PR(1)
50 FOR I=0 TO 5:FOR J=0 TO 5:
   LOCATE I*4+1,J*3+1:PRINT M0$:
   LOCATE I*4+1,J*3+2:PRINT M1$
60 A=RND(18):IF PT(A)=2 THEN 60
70 D(I,J)=A:PT(A)=PT(A)+1:NEXT:NEXT
80 PALETS 0,13,&H12,&H22,2:PALETS 1,13,&H14,&H24,4:
   PALETS 2,13,&H16,&H26,6:PALETS 3,13,2,25,&H36
90 DEF SPRITE 0,(3,1,0,0,0)=N$:DEF SPRITE 5,(3,1,0,0,0)=N$
100 LOCATE 25,9:PRINT"ﾋﾀﾞﾘ":LOCATE 25,12:PRINT"ﾐｷﾞ"
110 SPRITE 0,CX*32+24,CY*24+31:T0=-1:T1=-1:TT=7:GOSUB 170:
    SWAP T0,T1:X=CX:Y=CY:TT=6:GOSUB 170
120 IF T0<>T1 THEN 150
130 PLAY"T2O4C1E1G1O5C6":ED=ED+1:IF ED=18 THEN 300
140 PR(P)=PR(P)+10:LOCATE 23,10+P*3:PRINT PR(P):SPRITE 6:
    SPRITE 7:GOTO 110
150 PLAY"O1T2D5E5C5"
160 D(X,Y)=T1:D(CX,CY)=T0:LOCATE CX*4+1,CY*3+1:PRINT M0$:
    LOCATE CX*4+1,CY*3+2:PRINT M1$:LOCATE X*4+1,Y*3+1:PRINT M0$:
    LOCATE X*4+1,Y*3+2:PRINT M1$:P=1-P:GOTO 110
170 SPRITE 5,216,95+24*P:PAUSE 10:SPRITE 5:S=STICK(P):T=STRIG(P)
180 IF S=1 THEN CX=CX+1:IF CX>5 THEN CX=0
190 IF S=2 THEN CX=CX-1:IF CX<0 THEN CX=5
200 IF S=4 THEN CY=CY+1:IF CY>5 THEN CY=0
210 IF S=8 THEN CY=CY-1:IF CY<0 THEN CY=5
220 IF S=0 THEN 240
230 SPRITE 0,CX*32+24,CY*24+31
240 IF T<8 THEN 170
250 SWAP D(CX,CY),T0
260 IF T0=-1 THEN SWAP D(CX,CY),T0:GOTO 170
270 SPRITE 0:PLAY"O5T1C1C1":PAUSE 10:LOCATE CX*4+1,CY*3+1:
    PRINT"  ":LOCATE CX*4+1,CY*3+2:PRINT"  "
280 DEF SPRITE TT,(T0/6,1,1,0,0)=CHR$(48+T0-T0/6*6):
    SPRITE TT,CX*32+24,CY*24+31
290 RETURN
300 PRINT"TRY AGAIN !!"
310 T=STRIG(0):IF T=0 THEN 310
320 IF T=1 THEN RUN
330 IF T=2 THEN END
340 GOTO 300
350 END
```

〈CARD〉

トランプの神経衰弱の要領です。めくられたカードの数字と色を全部覚えておくか、それとも第六感で勝負するか。君は、どっちのタイプかな？

遊びかた

2人で遊びます。コントローラの✚ボタンでカードを選んで、Aボタンでめくります。2枚めくって同じ色の数字だったら、アタリ。続けてカードをめくることができます。違っていたら相手の番です。



★プログラムの変更や修正をする時の注意

- BASICプログラムの作成や変更および修正は、必ずBG GRAPHIC(背景)画面を消してから行なってください。
  背景の絵を出したままプログラムの変更や修正をすると、誤った結果となる場合があります。
- BG GRAPHIC画面を消すには SHIFT キーを押しながら CLR HOME キーを押してください。カーソルがホーム位置に戻ります。
- LISTでプログラムを呼びだし、変更や修正を行なってください。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02				
1	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
2	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
3	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32				
4	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
5	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
6	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32				
7	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
8	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
9	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	F72			
10	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
11	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
12	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	F72			
13	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
14	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
15	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32				
16	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
17	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
18	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22				
19																												
20																												

MEMO

MEMO

付　録

コントロール　コード

NS−HUBASICは、通常のオペレーションキーのほか、さらに扱いやすいように数多くのキー機能を備えています。

CTR+	処　理　内　容	参　考
A	[INS]モードのON／OFFスイッチ。	
C	BREAK。ただし、プログラム実行中はきかない。	
D	初期設定　※CGEN2、SPRITE　OFF、[CTR]+[A]の解除。カラーパレットをスプライト、バックグラウンド共に、バックグラウンド用のパレットコード1にする。	
E	カーソル以降1行分の消去。	
G	BEEP音を出す。	
H	[DEL]と同じ機能。	[DEL]
J	ラインフィード。(行送り)をする。[▼]キーと同じ。	
K	カーソルをホーム位置に戻す。	[SHIFT]+[CLR HOME]
L	画面のクリア。	
M	1行入力して改行。(キャリッジリターン)	[RETURN]
R	[INS]と同じ機能。	[INS]
V	カナモードにする。	
W	英・数字状態にする。	
Z	カーソル以降画面下までのクリア。	

※CTR+Aは[CTR]キーを押しながら[A]キーを押すことを意味しています。

メモリマップ

(16進表現)		
&H0000〜&H07FF	ワークRAM (ファミリーコンピュータ本体内)	
&H0800〜&H1FFF	未　使　用	
&H2000〜&H5FFF	システムで使用	
&H6000〜&H6FFF	未　使　用	
&H7000〜&H703F		&H7000〜&H703FのエリアはPOKE命令では使用しないでください。システムで使用しています。
〜&H77FF	ワークRAM (BASICカセット内)用 各種ボードデータまたはBASICフリーエリア	
&H7800〜&H7FFF	未　使　用	
&H8000〜&HFFFF	プログラムROM	

エラーメッセージ 一覧表

NS−HUBASICは、プログラム実行中にエラーを見つけると、エラーメッセージを画面に表示し、プログラムの実行を中止してコマンド(命令)の入力待ちの状態になります。

ダイレクトモードで実行中は、

(例えば)　?SN(シンタックス)　ERROR

プログラムモードで実行中は、

(例えば)　?SN　ERROR　IN　100

SN…エラーメッセージ
100…エラーとなった行番号

と表示されます。

もし、プログラム実行中に、エラーメッセージが表示され、エラーの原因がわからないときは、下記の一覧表を参照してください。

エラー表示	エラーメッセージ	説　明
NF	NEXT without FOR	FORがないのにNEXTがある。
SN	Syntax error	文法がまちがっている。
RG	RETURN without GOSUB	GOSUBがないのにRETURNがある。
OD	Out of DATA	READで読むべきデータがDATA文に用意されていない。
IL	Illegal function call	ステートメントや関数の呼び方が間違っている。
OV	Overflow	演算結果が許容範囲を越えた。
OM	Out of memory	メモリ不足である。
UL	Undefined line Number	GOTO、GOSUB、IFなどで指定した分岐先の行番号がない。
SO	Subscript out of range	配列変数の添字が規定外である。
DD	Duplicate Definition	配列が2重に定義されている。
DZ	Division by zero	0による除算をした。
TM	Type mismatch	変数の型が一致しない。
ST	String too long	文字が31文字を越えている。
FT	Formula Too complex	式が複雑すぎる。たとえば、( )が異常に多い場合。
CC	Can't continue	CONTによってプログラムの実行を再開できない。
MO	Missing operand	パラメータの必要な命令に指定がない。
TP	Tape read ERROR	カセットテープからデータが正しく読めない。

キャラクタコード表

ファミリーベーシックのキャラクタコードを、一覧表で示します。BASICのCHR$（n）、ASC（"A$"）による文字とコードの変換は、これに対応しています。キャラクタコード表A、またはBのキャラクタ、文字、記号は、バックグラウンド、スプライトの各面で使用できます。（どの面にどちらの表を使用するかの指定はCGEN命令を使用します）

●キャラクタコード表A（主に、スプライト用）

コード（10進数）は、キャラクタテーブルAでアニメキャラクタの四隅に表記してある番号に対応します。

コード（10進数）	コード（16進数）	説明	コード（10進数）	コード（16進数）	説明	コード（10進数）	コード（16進数）	説明	コード（10進数）	コード（16進数）	説明
0	00	マリオ（WALK1）	32	20	レディ（WALK2）	64	40	アキレス（左1）	96	60	ペンペン（左歩1）
1	01		33	21		65	41		97	61	
2	02		34	22		66	42		98	62	
3	03		35	23		67	43		99	63	
4	04	マリオ（WALK2）	36	24	レディ（WALK3）	68	44	アキレス（左2）	100	64	ペンペン（左歩2）
5	05		37	25		69	45		101	65	
6	06		38	26		70	46		102	66	
7	07		39	27		71	47		103	67	
8	08	マリオ（WALK3）	40	28	レディ（JUMP）	72	48	アキレス（左上1）	104	68	ペンペン（正面）
9	09		41	29		73	49		105	69	
10	0A		42	2A		74	4A		106	6A	
11	0B		43	2B		75	4B		107	6B	
12	0C	マリオ（JUMP）	44	2C	レディ（スリップ着地）	76	4C	アキレス（左上2）	108	6C	ペンペン（後）
13	0D		45	2D		77	4D		109	6D	
14	0E		46	2E		78	4E		110	6E	
15	0F		47	2F		79	4F		111	6F	
16	10	マリオ（スリップ着地）	48	30	レディ（はしご）	80	40	アキレス（上1）	112	70	ファイアーボール(1)
17	11		49	31		81	51		113	71	
18	12		50	32		82	52		114	72	
19	13		51	33		83	53		115	73	
20	14	マリオ（はしご）	52	34	レディ（DOWN）	84	54	アキレス（上2）	116	74	ファイアーボール(2)
21	15		53	35		85	55		117	75	
22	16		54	36		86	56		118	76	
23	17		55	37		87	57		119	77	
24	18	マリオ（DOWN）	56	38	ファイターフライ(1)	88	58	ニタニタ(1)	120	78	車（左1）
25	19		57	39		89	59		121	79	
26	1A		58	3A		90	5A		122	7A	
27	1B		59	3B		91	5B		123	7B	
28	1C	レディ（WALK1）	60	3C	ファイターフライ(2)	92	5C	ニタニタ(2)	124	7C	車（左2）
29	1D		61	3D		93	5D		125	7D	
30	1E		62	3E		94	5E		126	7E	
31	1F		63	3F		95	5F		127	7F	

●アニメキャラクタを定義する場合は、このコード表にあるコード（10進数、または16進数）で指定してください。
●∅～255までの10進と16進の換算表としてもご利用ください。

コード（10進数）	コード（16進数）	説明	コード（10進数）	コード（16進数）	説明	コード（10進数）	コード（16進数）	説明	コード（10進数）	コード（16進数）	説明
128	80	車（左上1）	160	A0	スターキラー（上）	192	C0	サイドステッパー(1)	224	E0	はねペン(1)
129	81		161	A1		193	C1		225	E1	
130	82		162	A2		194	C2		226	E2	
131	83		163	A3		195	C3		227	E3	
132	84	車（左上2）	164	A4	スターシップ（左）	196	C4	サイドステッパー(2)	228	E4	
133	85		165	A5		197	C5		229	E5	
134	86		166	A6		198	C6		230	E6	
135	87		167	A7		199	C7		231	E7	
136	88	車（上1）	168	A8	スターシップ（左上）	200	C8	ニットピッカー(1)	232	E8	はねペン(2)
137	89		169	A9		201	C9		233	E9	
138	8A		170	AA		202	CA		234	EA	
139	8B		171	AB		203	CB		235	EB	
140	8C	車（上2）	172	AC	スターシップ（上）	204	CC	ニットピッカー(2)	236	EC	
141	8D		173	AD		205	CD		237	ED	
142	8E		174	AE		206	CE		238	EE	
143	8F		175	AF		207	CF		239	EF	
144	90	スピナー(1)	176	B0	爆発(1)	208	D0	レーザー（よこ）	240	F0	1
145	91		177	B1		209	D1		241	F1	2
146	92		178	B2		210	D2	レーザー（ななめ）	242	F2	3
147	93		179	B3		211	D3		243	F3	4
148	94	スピナー(2)	180	B4	爆発(2)	212	D4	レーザー（たて）	244	F4	COMPUTERの文字
149	95		181	B5		213	D5		245	F5	
150	96		182	B6		214	D6	スプライト用ぬりつぶしの色(1)、(2)、(3)	246	F6	
151	97		183	B7		215	D7		247	F7	
152	98	スターキラー（左）	184	B8	シェルクリーパー(1)	216	D8		248	F8	
153	99		185	B9		217	D9	ミュージックボード用カーソル(1)、(2)、(3)	249	F9	
154	9A		186	BA		218	DA		250	FA	OPERATORの文字
155	9B		187	BB		219	DB		251	FB	
156	9C	スターキラー（左上）	188	BC	シェルクリーパー(2)	220	DC	第1カーソル	252	FC	
157	9D		189	BD		221	DD	第2カーソル	253	FD	
158	9E		190	BE		222	DE	ランプ(1)	254	FE	
159	9F		191	BF		223	DF	ランプ(2)	255	FF	

●キャラクタコード表B（主に、バックグラウンド面用、及びキーボードの文字、記号、BG GRAPHICに使用できるキャラクタです。）

対応する文字の欄に（　）で書いてある番号は、キャラクタテーブルBのキャラクタに対応します。（113ページ）

コード（10進数）	コード（16進数）	対応する文字	コード（10進数）	コード（16進数）	対応する文字	コード（10進数）	コード（16進数）	対応する文字	コード（10進数）	コード（16進数）	対応する文字
0	00	(A0)	32	20		64	40	@	96	60	ア
1	01	(A1)	33	21	!	65	41	A	97	61	イ
2	02	(A2)	34	22	”	66	42	B	98	62	ウ
3	03	(A3)	35	23	#	67	43	C	99	63	エ
4	04	(A4)	36	24	$	68	44	D	100	64	オ
5	05	(A5)	37	25	%	69	45	E	101	65	カ
6	06	(A6)	38	26	&	70	46	F	102	66	キ
7	07	(A7)	39	27	’	71	47	G	103	67	ク
8	08	(B0)	40	28	(	72	48	H	104	68	ケ
9	09	(B1)	41	29	)	73	49	I	105	69	コ
10	0A	(B2)	42	2A	*	74	4A	J	106	6A	サ
11	0B	(B3)	43	2B	+	75	4B	K	107	6B	シ
12	0C	(B4)	44	2C	,	76	4C	L	108	6C	ス
13	0D	(B5)	45	2D	−	77	4D	M	109	6D	セ
14	0E	(B6)	46	2E	.	78	4E	N	110	6E	ソ
15	0F	(B7)	47	2F	／	79	4F	O	111	6F	タ
16	10	(C0)	48	30	0	80	50	P	112	70	チ
17	11	(C1)	49	31	1	81	51	Q	113	71	ツ
18	12	(C2)	50	32	2	82	52	R	114	72	テ
19	13	(C3)	51	33	3	83	53	S	115	73	ト
20	14	(C4)	52	34	4	84	54	T	116	74	ナ
21	15	(C5)	53	35	5	85	55	U	117	75	ニ
22	16	(C6)	54	36	6	86	56	V	118	76	ヌ
23	17	(C7)	55	37	7	87	57	W	119	77	ネ
24	18	(D0)	56	38	8	88	58	X	120	78	ノ
25	19	(D1)	57	39	9	89	59	Y	121	79	ハ
26	1A	(D2)	58	3A	:	90	5A	Z	122	7A	ヒ
27	1B	(D3)	59	3B	;	91	5B	「	123	7B	フ
28	1C	(D4)	60	3C	<	92	5C	¥	124	7C	ヘ
29	1D	(D5)	61	3D	=	93	5D	」	125	7D	ホ
30	1E	(D6)	62	3E	>	94	5E	^	126	7E	マ
31	1F	(D7)	63	3F	?	95	5F	−	127	7F	ミ

●バックグラウンド用に直接キャラクタを定義する場合は、このコード表にあるコード（10進数、または16進数）で指定してください。
ただし、コード(16進数)00～1Fは、システムで使用しているため直接指定することはできません。
●0～255までの10進と16進の換算表としてもご利用ください。

コード(10進数)	コード(16進数)	対応する文字
128	80	ム
129	81	メ
130	82	モ
131	83	ヤ
132	84	ユ
133	85	ヨ
134	86	ラ
135	87	リ
136	88	ル
137	89	レ
138	8A	ロ
139	8B	ワ
140	8C	ン
141	8D	ヲ
142	8E	ァ
143	8F	ィ
144	90	ゥ
145	91	ェ
146	92	ォ
147	93	ャ
148	94	ュ
149	95	ョ
150	96	ッ
151	97	ガ
152	98	ギ
153	99	グ
154	9A	ゲ
155	9B	ゴ
156	9C	ザ
157	9D	ジ
158	9E	ズ
159	9F	ゼ

コード(10進数)	コード(16進数)	対応する文字
160	A0	ゾ
161	A1	ダ
162	A2	ヂ
163	A3	ヅ
164	A4	デ
165	A5	ド
166	A6	バ
167	A7	ビ
168	A8	ブ
169	A9	ベ
170	AA	ボ
171	AB	パ
172	AC	ピ
173	AD	プ
174	AE	ペ
175	AF	ポ
176	B0	□
177	B1	°
178	B2	[
179	B3	]
180	B4	©
181	B5	×
182	B6	÷
183	B7	ファ
184	B8	(E0)
185	B9	(E1)
186	BA	(E2)
187	BB	(E3)
188	BC	(E4)
189	BD	(E5)
190	BE	(E6)
191	BF	(E7)

コード(10進数)	コード(16進数)	対応する文字	
192	C0	(F0)	マリオ用
193	C1	(F1)	
194	C2	(F2)	
195	C3	(F3)	
196	C4	(F4)	
197	C5	(F5)	
198	C6	(F6)	
199	C7	(F7)	
200	C8	(G0)	LANDSCAPE
201	C9	(G1)	
202	CA	(G2)	
203	CB	(G3)	
204	CC	(G4)	
205	CD	(G5)星(1)	
206	CE	(G6)星(2)	
207	CF	(G7)ボール	
208	D0	(H0)	ドンキーコングJr.用
209	D1	(H1)	
210	D2	(H2)	
211	D3	(H3)	
212	D4	(H4)	
213	D5	(H5)	
214	D6	(H6)	
215	D7	(H7)	
216	D8	(I0)	島
217	D9	(I1)	
218	DA	(I2)	
219	DB	(I3)	
220	DC	(I4)	
221	DD	(I5)	迷路
222	DE	(I6)	
223	DF	(I7)	

コード(10進数)	コード(16進数)	対応する文字	
224	E0	(J0)	迷路
225	E1	(J1)	
226	E2	(J2)	
227	E3	(J3)	
228	E4	(J4)	
229	E5	(J5)	
230	E6	(J6)	
231	E7	(J7)	
232	E8	(K0)	グラフ用ケイ線
233	E9	(K1)	
234	EA	(K2)	
235	EB	(K3)	
236	EC	(K4)	
237	ED	(K5)	
238	EE	(K6)	
239	EF	(K7)	
240	F0	(L0)	
241	F1	(L1)	
242	F2	(L2)	
243	F3	(L3)	ボード用ワク
244	F4	(L4)	
245	F5	(L5)	
246	F6	(L6)	
247	F7	(L7)	
248	F8	(M0)	インクツボ
249	F9	(M1)	
250	FA	(M2)	
251	FB	(M3)	
252	FC	(M4)	
253	FD	(M5)(1)	ヌリツブシ
254	FE	(M6)(2)	
255	FF	(M7)(3)	

アルファベット順命令語索引

〔　〕内は各命令語の省略形を示しています。

お取扱い上のご注意

接続コードについて	キーボードの接続コードを机やイスの下に敷いたり、ものにはさんで傷をつけないようご注意ください。接続コードに傷がついたまま使用すると危険です。また接続コードを抜くときはかならずコネクタを持って抜いてください。
電源電圧について	ファミリーコンピュータの電源電圧はAC100Vでご使用ください。電源電圧が極端に高かったり、低かったりすると故障の原因になり、十分性能が発揮できない場合があります。このようなときはお買い上げの販売店あるいはもよりの任天堂㈱、各支店・営業所にご相談ください。
湿気やほこりについて	本機を湿気の多い場所や、ほこりの多い場所に置かないでください。故障の原因になります。また、ほこりを吸い込まぬ様ほこりの多い場所での使用はさけてください。
高温について	本機を直射日光の当る場所や暖房器具の近くに置かないでください。キャビネットや内部の部品をいためる原因になります。
水や異物について	本機の内部に水や液状のもの、針やピンなど金属類が入ったまま使用すると危険です。異物が入らないようご注意ください。とくに水や液状の異物が入った場合すぐにキーボードのコネクタを抜き、お買い上げの販売店にご連絡ください。
衝撃について	本機は精密な電子部品でできています。落としたり、物を当てたりして衝撃を与えないでください。故障の原因になります。
異常のときは	故障した時や異常を感じたら、使用を中止し、お買い上げの販売店あるいはもよりの任天堂(株)各支店・営業所にご相談ください。
長時間使用しない場合	長い間ご使用にならない場合はかならずファミリーコンピュータからBASIC カセット、及びキーボードのコネクタを抜いてください。
周辺機器の増設	モノラルラジカセ以外の使用は故障の原因になる恐れがあります。
汚れ	本機の汚れはやわらかい布に水または洗剤を含ませて軽くふいてください。ベンジン、シンナーなど揮発性のものは使用しないでください。キャビネットの変色などの原因になります。
電源の「入－切」	ファミリーコンピュータの電源スイッチの「入－切」は10秒以上の間隔をあけて操作してください。コンピュータの動作を確実にするため必要です。またスイッチの入った状態でBASICカセットやキーボードのコネクタを抜いたり差したりすると、故障の原因になります。

乾電池お取扱い上のご注意

メモリバックアップのためには、単三乾電池２本が必要です。乾電池は誤った使い方をしますと液もれや破れつする危険がありますので、次の点にご注意ください。

1．乾電池のプラス⊕とマイナス⊖の向きをBASICカセットの表示どおり正しく入れてください。
2．新しい乾電池と一度使用した乾電池を混ぜて使用しないでください。
3．種類のちがう乾電池を混ぜて使用しないでください。同じ形状でも電圧の異なるものがあります。
4．乾電池は使えなくなったり、長時間使わないときはBASICカセットから取りはずしておいてください。

（キリトリ線）

ファミリーベーシック™ 保証書

お客様	ご住所 〒 TEL. お名前 59.7.14	お買い上げ店	住所 トロイの木馬 小金井市東町4-42-20 TEL 0423-85-0366 店名 ㊞	
お買い上げ月日	年　月　日	機種 HVC-007	製造番号	

● 正常な使用状態のもとで発生した故障については、**お買い上げの日から６ヵ月**の期間、裏面記載の保証規定にもとづき、無償修理いたします。
● 故障が生じました場合は、お買い求めのお店にご連絡ください。なお、本保証書にお買い上げ店・お買い上げ年月日が記入されているかをお確かめください。本書は再発行いたしません。
※この保証書は〝カセット〟の保証を含みません。

任天堂株式会社

（キリトリ線）

保証書およびサービスについて

★保証書について
保証書は、このたびお買い上げいただきました任天堂製品の品質を保証するもので故障の場合の、保証期間内無料サービスに必要です。もし未発行の保証書をお持ちの場合は、保証書の該当欄に、お名前、ご住所を記入の上、お買い上げの販売店にて発行手続きをお受けください。（保証に関する事項は保証書に記載しています。）　※この保証書は〝カセット〟の保証を含みません。

★サービスを依頼されるときは……
故障が生じました場合は、お買い上げ店にご連絡ください。
なお、保証書には、お買い上げ店・お買い上げ年月日が記入されているかをお確かめください。

ファミリーベーシックの仕様

●ソフト：1．BASIC（BG GRAPHICを含む）
2．カリキュレータボード
3．メッセージボード
4．コンピュータのうらない
5．ミュージックボード

項目		仕様
ROM		プログラム用ROM……8ビット×16K×2個 キャラクタゼネレータ用ROM……8ビット×8K
RAM		ワーキング及びメモリバックアップ用　S−RAM……8ビット×2K
表示能力	文字表示	28文字×24行（BASIC時）
	文字構成	8×8ドットマトリックス文字　英文字、数字、カタカナ、英記号、特殊記号、キャラクタ
	表現色数	52色（白及、黒を含む）
	動画（スプライト）	8スプライト（但し、1スプライト　16×16ドットまたは、8×8ドット） 256×240ドット（BASIC時）
サウンド		2個の方形波音 1個の三角波音 1個の効果音
キーボード	キーの数	総数72キー
	配　列	ASCⅡ配列（但し、カタカナ50音配列）準拠
	ファンクションキー	8キー
	カーソルキー	4キー
インターフェイス		1200ボー　カセットインターフェイス内蔵 3.5φ READ、WRITE端子付
メモリバックアップ		BASICカセット内に、単三乾電池を内蔵する事によりBASICプログラムや、各ボード実行データを、一時的に記憶させておく事ができる。
寸法	キーボード（HVC-007）	368（W）×183（D）×54（H）
	カセット　（HVC-FB）	110（W）×110（D）×17（H）
重量	キーボード（HVC-007）	956g
	カセット　（HVC-FB）	95g（電池含まず）
使用条件		使用温度0〜40℃、使用湿度35〜75％
附属品		キャラクタマップ

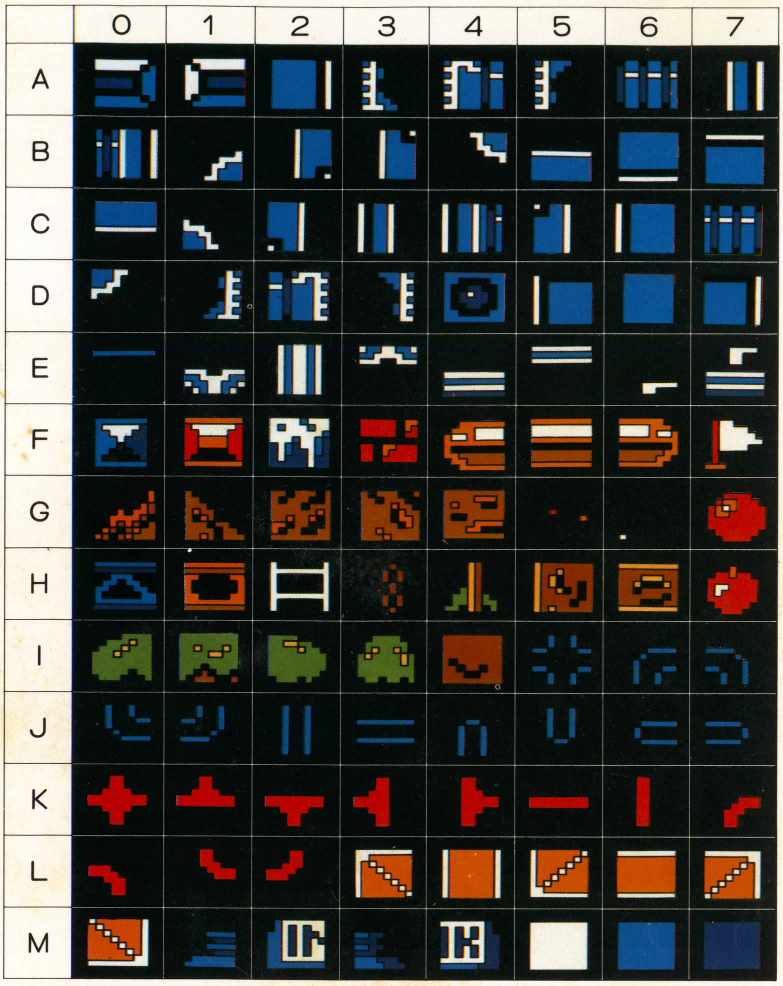
※仕様および外観の一部を予告なく変更する場合がありますのでご了承ください。

保証規定

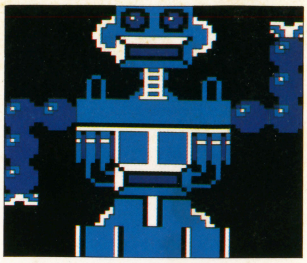
1．表面記載の保証期間内に発生した故障については修理料金を無償といたします。
2．次の場合は保証期間中でも有償修理といたします。
①使用上の誤り、および不当な修理や改造による故障。
②火災・天災などによる故障および購入後の輸送による故障。
③接続している他の機器に起因して本製品に生じる故障。
④本保証書を紛失された場合、およびお買い上げ日、お買い上げ店・認印のない場合。
⑤本保証書を弊社に提示されていない場合。
3．この保証書は国内で使用される場合にだけ有効です。
This warranty shall be valid only within Japan.

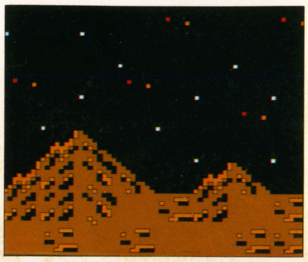
本　　社　〒605 京都市東山区福稲上高松町60番地　TEL（075）541−6111番（代表）
東京支店　〒101 東京都千代田区神田須田町1丁目22　TEL（03）254−1781番（代表）
大阪支店　〒542 大阪市南区長堀橋筋1丁目32　TEL（06）271−5514番（代表）
名古屋営業所　〒451 名古屋市西区幅下町2丁目18番9号　TEL（052）571−2506番（代表）
札幌営業所　〒060 札幌市中央区北九条西18丁目2番地　TEL（011）621−0513番（代表）
岡山営業所　〒700 岡山市奉還町4丁目4番11号　TEL（0862）52−1821番（代表）

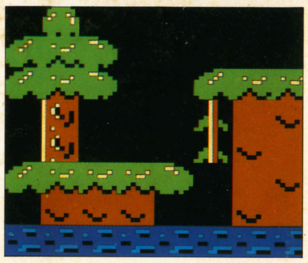
キャラクタテーブルB



「BG GRAPHIC」のキャラクタを利用して、いろいろな絵が描けます。右図はその例です。なお、これらは64×64ドットで表わしたものであり、画面全体ではありません。


ロボット


山と星


木と島

キャラクタテーブルBに記載しているキャラクタは、「BG GRAPHIC」のSELECTモードで表示されるキャラクタグループを表わしています。8個のキャラクタグループを変えるには、[CLR HOME]キー、または、[SHIFT]キーを押しながら[CLR HOME]キーを押します。

カラーチャート

カラーチャートは、CGSET文で指定される色の組み合わせを表わしています。この配色を参考にして、キャラクタの色を指定してください。

スプライト用

CGSETで指定されるパレットコード

DEF SPRITE、DEF MOVE命令で指定される配色番号。

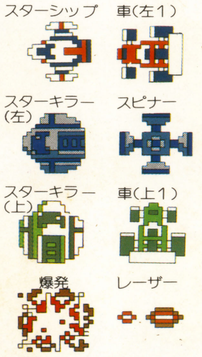
パレットコード0

配色番号	色1	色2	色3
0	54 / 36	22 / 16	2 / 02
1	39 / 27	48 / 30	25 / 19
2	53 / 35	37 / 25	23 / 17
3	48 / 30	39 / 27	22 / 16



パレットコード1

配色番号	色1	色2	色3
0	48 / 30	22 / 16	1 / 01
1	16 / 10	0 / 00	1 / 01
2	48 / 30	41 / 29	9 / 09
3	48 / 30	22 / 16	7 / 07



パレットコード2

配色番号	色1	色2	色3
0	48 / 30	38 / 26	18 / 12
1	48 / 30	21 / 15	18 / 12
2	48 / 30	18 / 12	22 / 16
3	48 / 30	38 / 26	25 / 19



バックグラウンド用

CGSETで指定されるパレットコード

COLOR命令で指定される配色番号。

パレットコード0

配色番号	色1	色2	色3
0	44 / 2C	21 / 15	7 / 07
1	39 / 27	33 / 21	18 / 12
2	41 / 29	54 / 36	23 / 17
3	48 / 30	38 / 26	7 / 07

パレットコード1

配色番号	色1	色2	色3
0	48 / 30	33 / 21	2 / 02
1	48 / 30	39 / 27	24 / 18
2	48 / 30	39 / 27	22 / 16
3	41 / 29	54 / 36	23 / 17

BG GRAPHICで表示される配色番号

各配色の下に表記してある数字(カラーコード)は、10進数(上段)、16進数(下段)の2種類です。PALET文でアニメキャラクタや背景の絵柄の色を変える場合、これらのカラーコードと対応させて色を指定変更してください。

※テレビにより、印刷の色と多少異なって表示される場合があります。

キャラクタテーブルA



これらのキャラクタは、主にアニメキャラクタとして使用します。
各キャラクタの四隅に表記している数字は、DEF SPRITE文のCHR$（n）で使用する10進数です。

任天堂株式会社

本社 〒605 京都市東山区福稲上高松町60番地 TEL（075）541－6111番（代表）
東京支店 〒101 東京都千代田区神田須田町1丁目22 TEL（03）254－1781番（代表）
大阪支店 〒542 大阪市南区長堀橋筋1丁目32 TEL（06）271－5514番（代表）
名古屋営業所 〒451 名古屋市西区幅下2丁目18番9号 TEL（052）571－2506番（代表）
札幌営業所 〒060 札幌市中央区北九条西18丁目2番地 TEL（011）621－0513番（代表）
岡山営業所 〒700 岡山市奉還町4丁目4番11号 TEL（0862）52－1821番（代表）

〔正誤表〕

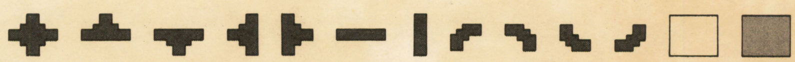
		誤	正
P㊴	Rock'n Rouge	（ロックル・ルージュ）	（ロックン・ルージュ）
P㊵	サンプルプログラム2 画面の上　3行目	行番号10	行番号20
P㊷	サンプルプログラム4 行番号390	PRTINT	PRINT

〔追　加〕

P⑨「メッセージボード画面」

●グラフィック記号の使い方

カナ状態で GRPH キーを押しながら文字キーの最上列を押すと下のグラフィック記号が描けます。



※「2. 手軽な記録板」の表は、この方法で描いています。